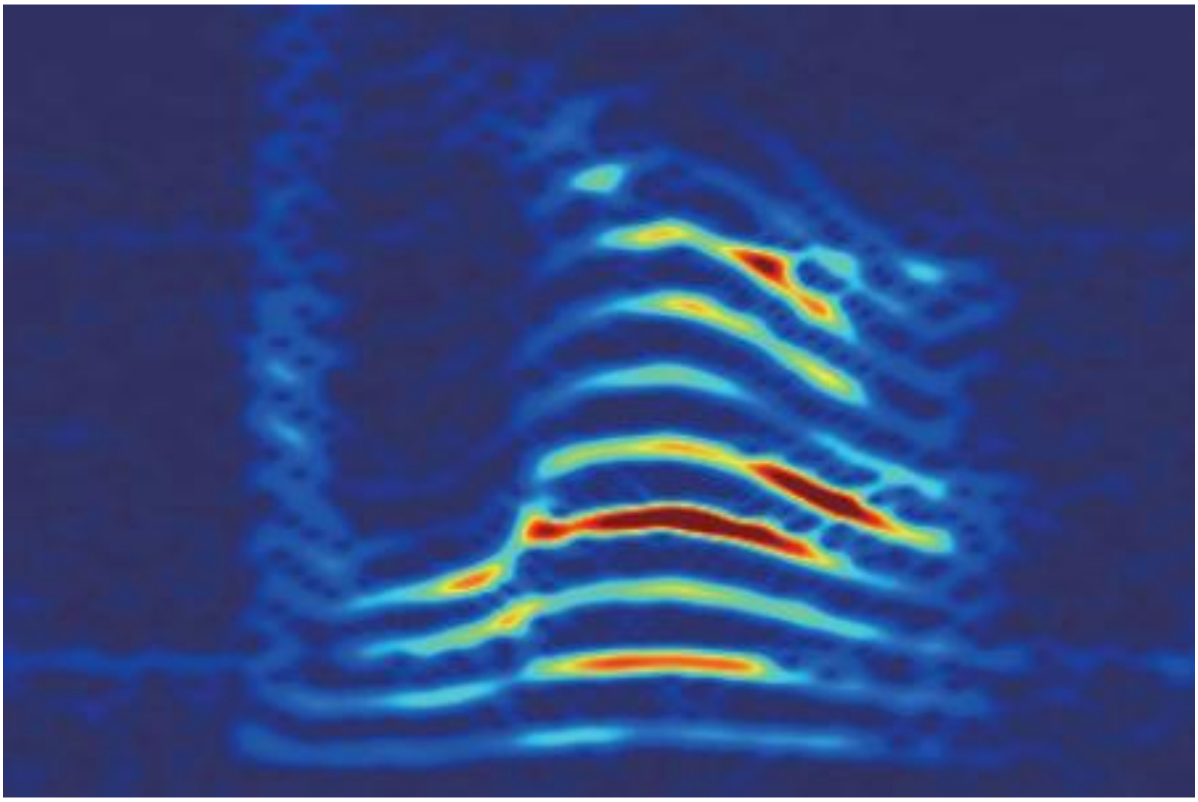
DataDemon User Manual

Version 1.4.0



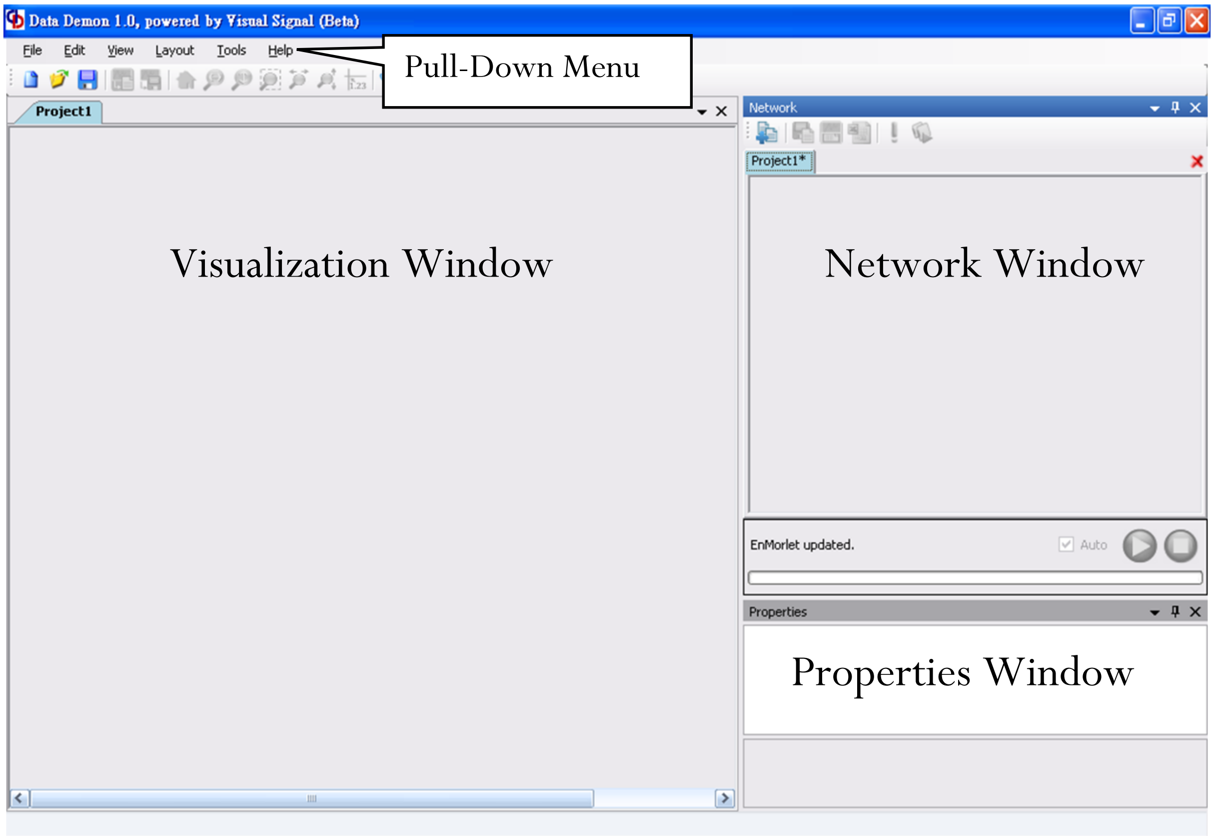
DataDemon User Manual

[Version 1.4.0]

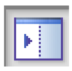
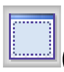
Chapter 1: User Interface

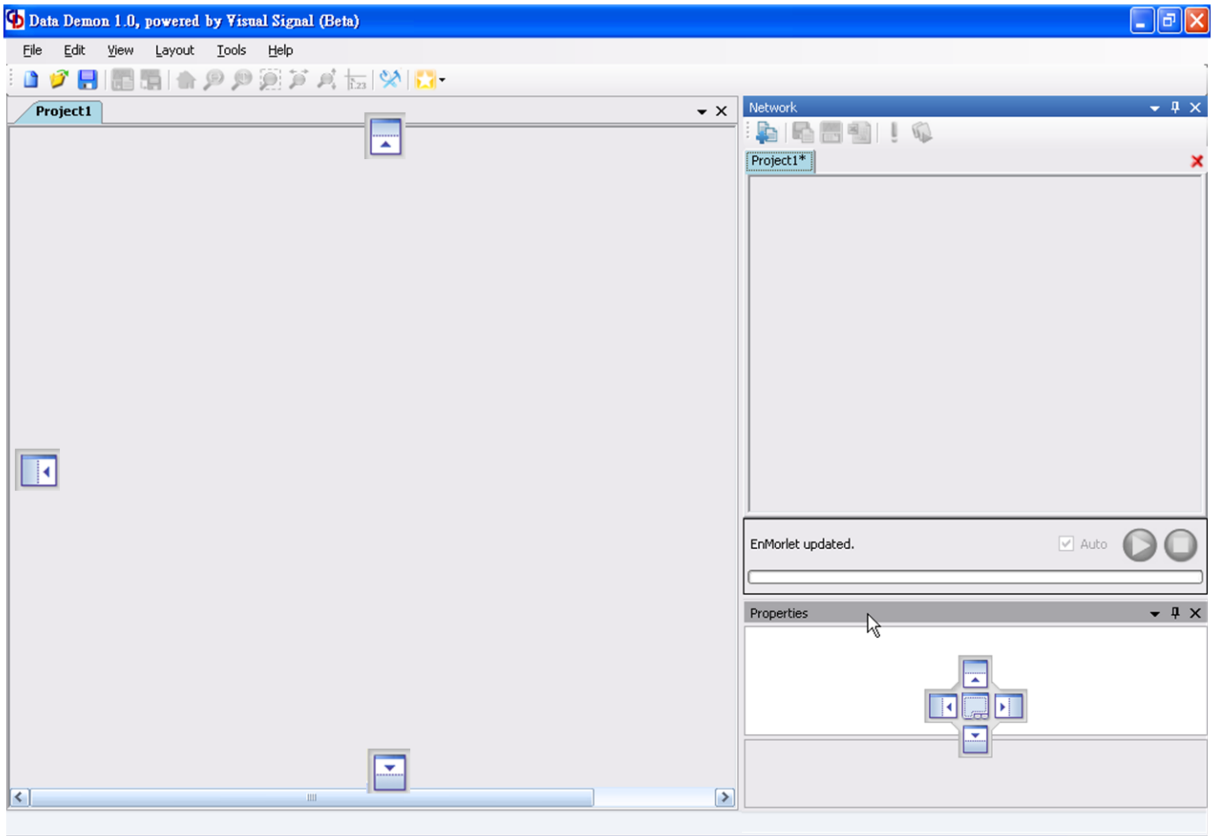
1.1 Introduction

This chapter introduces the layout and user interface of DataDemon version 1.4.0. The user interface (UI) of DataDemon is simple and easy to use and it utilizes many customizable characteristics to suit your personal preferences. The following image shows the UI of the DataDemon after the program is executed.

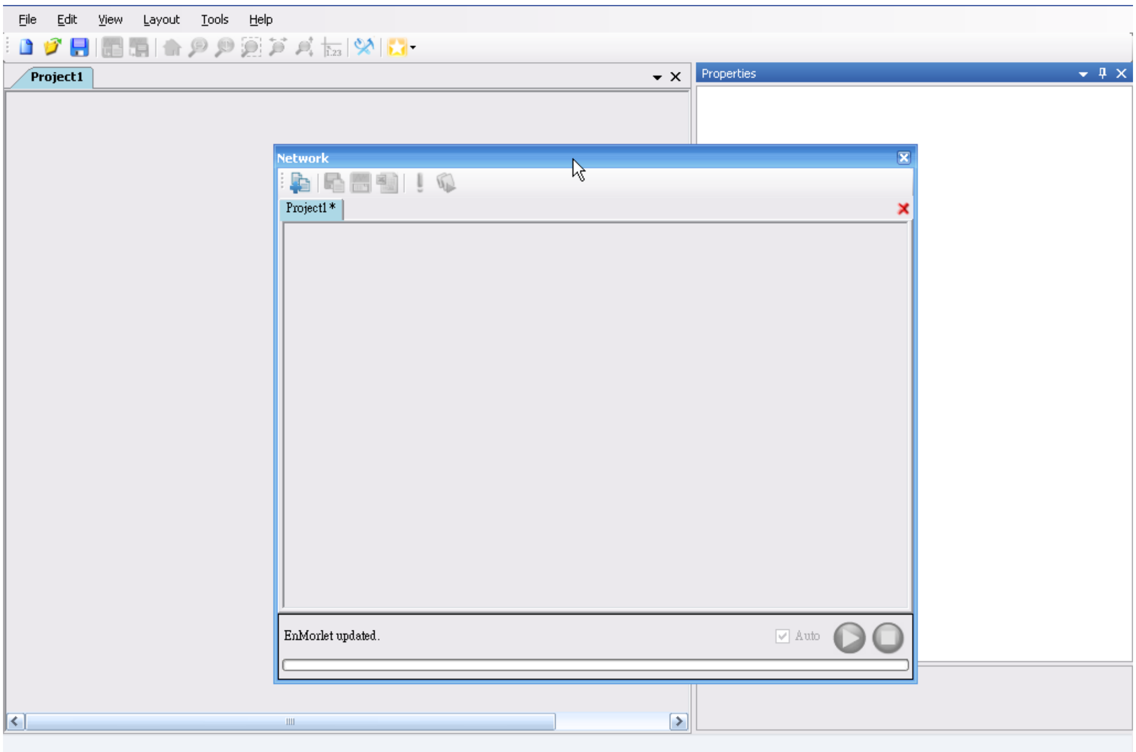


The UI (User Interface) is divided into three major sections: *network window*, *Properties Window* and *Visualization Window*.

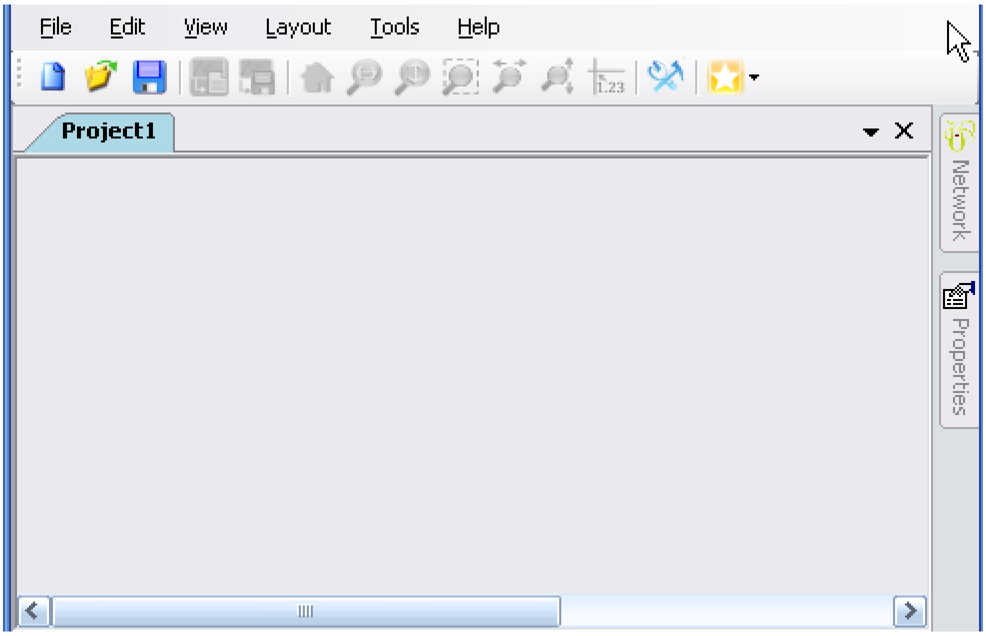
The positions of these three major sections can be customized. You can drag and drop each individual window onto any part of the UI or even combine the windows, so they can be viewed through the display tabs. To drag and drop a particular window, move the mouse pointer to the top bar of the display window and left click and hold down the mouse button to drag. During the process of drag and drop, position icons such as  (Top),  (Bottom),  (Left),  (Right), (Expand) and  (Tab) will appear. Drag and drop the mouse over to one of the position icons so the window will change according to the position icon selected.



Double click the top bar of a section display window to detach the window from its current position and it will appear in front of the other section windows (as shown in the image below). Double click the top bar of the section display window again to restore the detached window to its original position.



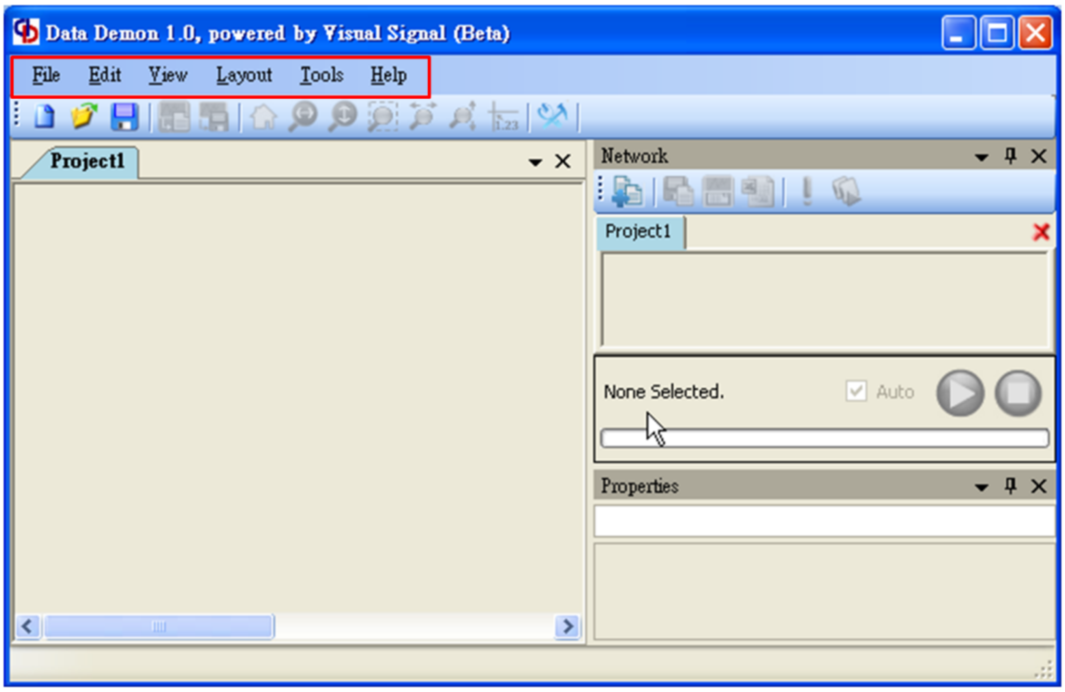
Section window also supports *auto hide* feature to automatically hide the window when it is not in use. Click on the icon in the Network and *Properties Window* to configure the window to auto hide to the side of the main window area (as shown in the image below).



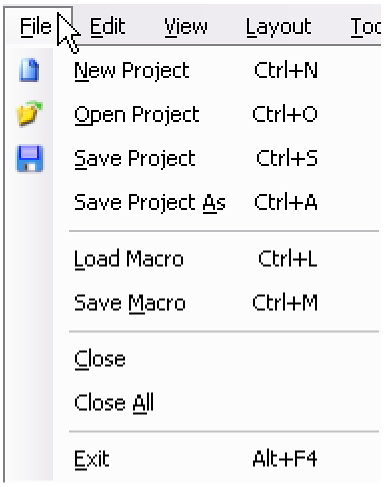
In next section, menu bar and other program functionalities will be explained.

1.2 Pull-down Menu

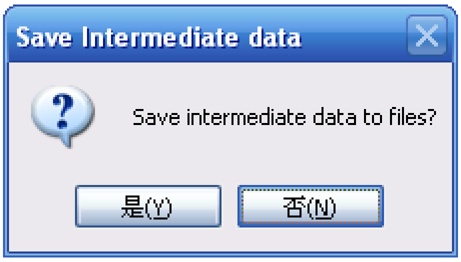
In the top left section of the program, there is the menu bar which consists of several *pull-down menus*: *File*, *Edit*, *View*, *Layout*, *Tools and* Help. The menu options such as *Edit*, *View* and *Tools* are used for editing and configuring *Visualization Window* and it will be explained in the *Visualization Window* chapter. Only the contents of *File*, *Layout* and *Help* will be explained in this chapter.



File allows you to create, open and save project files in the DataDemon format (.VSN).



When saving a project, you can choose whether to save the calculations during this project or not. If *Yes* is clicked, all calculations will be saved and they will be restored when reopened next time.



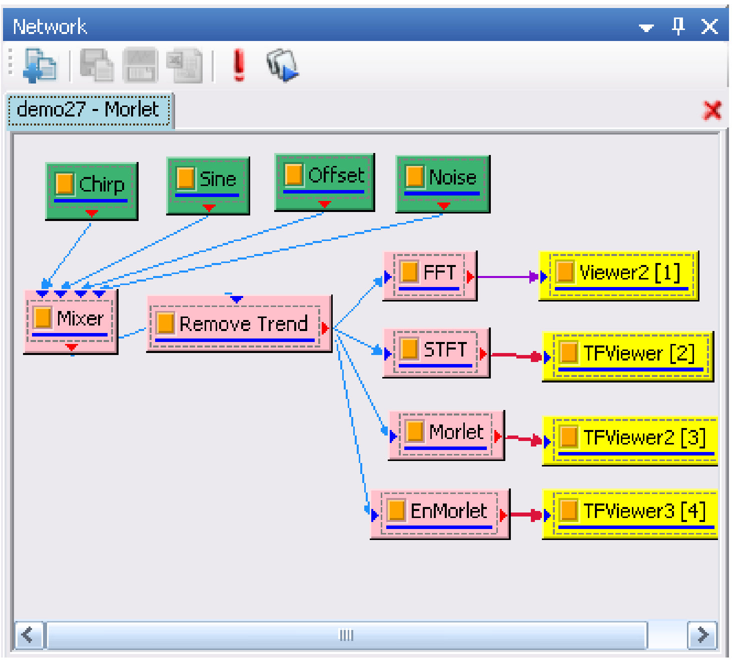
If *No* is clicked, when the project is opened next time, all of the calculations need to be processed again. Selecting *Close* will close the current working project or selecting *Close All* will close all projects currently opened in DataDemon.

Load Macro and *Save Macro* are new features only available in DataDemon Professional or above. The implementation of the macro will allow you to quickly save and load the Signal Flow Diagrams that have been created. The macro can be added to any project.

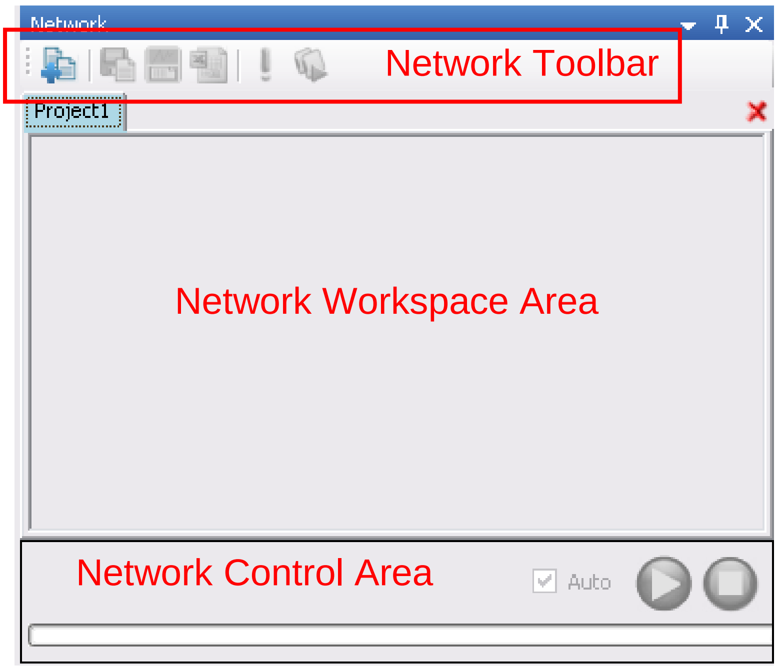
(**TIPS**: To get the feel of how to use the macro effectively, create a simple project and play around with saving and loading macro files).

1.3 Network Window

Network Window contains the calculation steps for the program. You can create, execute and link Signal Flow Objects (SFOs) by using the mouse to click on or drag the SFOs to setup basic Signal Flow Diagrams for data calculation and manipulation.

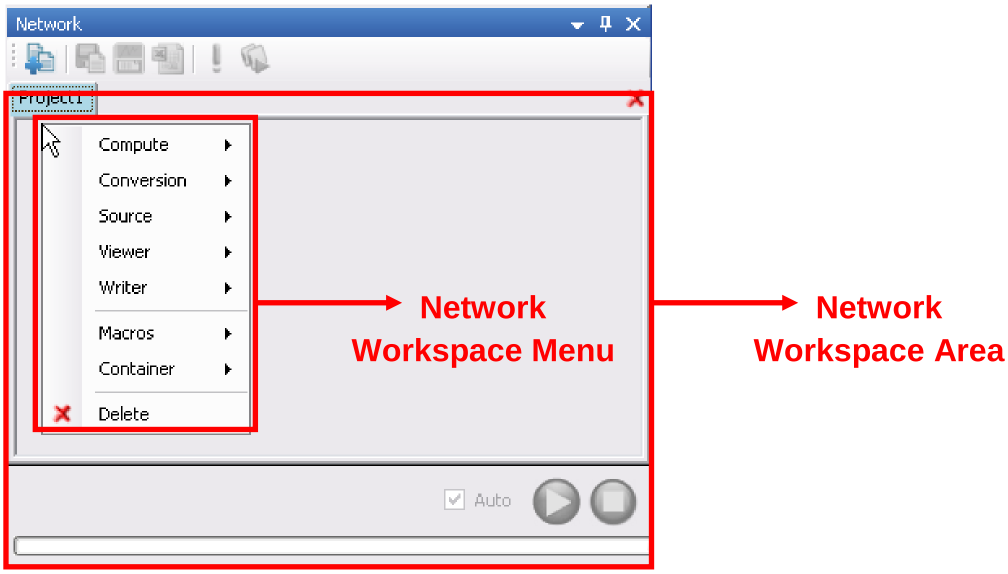


Network Window can be divided into three areas; *Network Window Toolbar, Network Workspace* and *Network Control Area*.

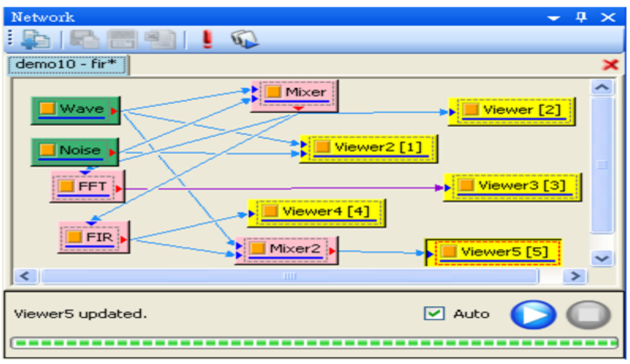


1.3.1 Network Workspace Area

Network Workspace Area is the core of DataDemon. DataDemon has implemented the object oriented design. Under DataDemon object oriented environment, it is simple to create and edit Signal Flow Objects and Signal Flow Diagrams in *Network Workspace Area*. And signals can be easily visualized through the Signal Flow Diagrams between Signal Flow Objects (SFOs).



Shown in the above image is the *Network Workspace Menu* that will appear once the right mouse is clicked within the *Network Workspace*. In the *Network Workspace Menu* contains five categories; *Compute, Conversion*, *Source*, *Viewer* and *Writer* which can be used to display the building blocks of the Signal Flow Diagrams created in the *Network Workspace*. The way to add and edit the five categories will be explained in Chapter 2 and details of the computational aspect of the signal processing and methods will be explained in chapter 3 and 4. In the *Network Workspace* Area, you can easily create, edit, process and view signal results through a few simple mouse clicks.

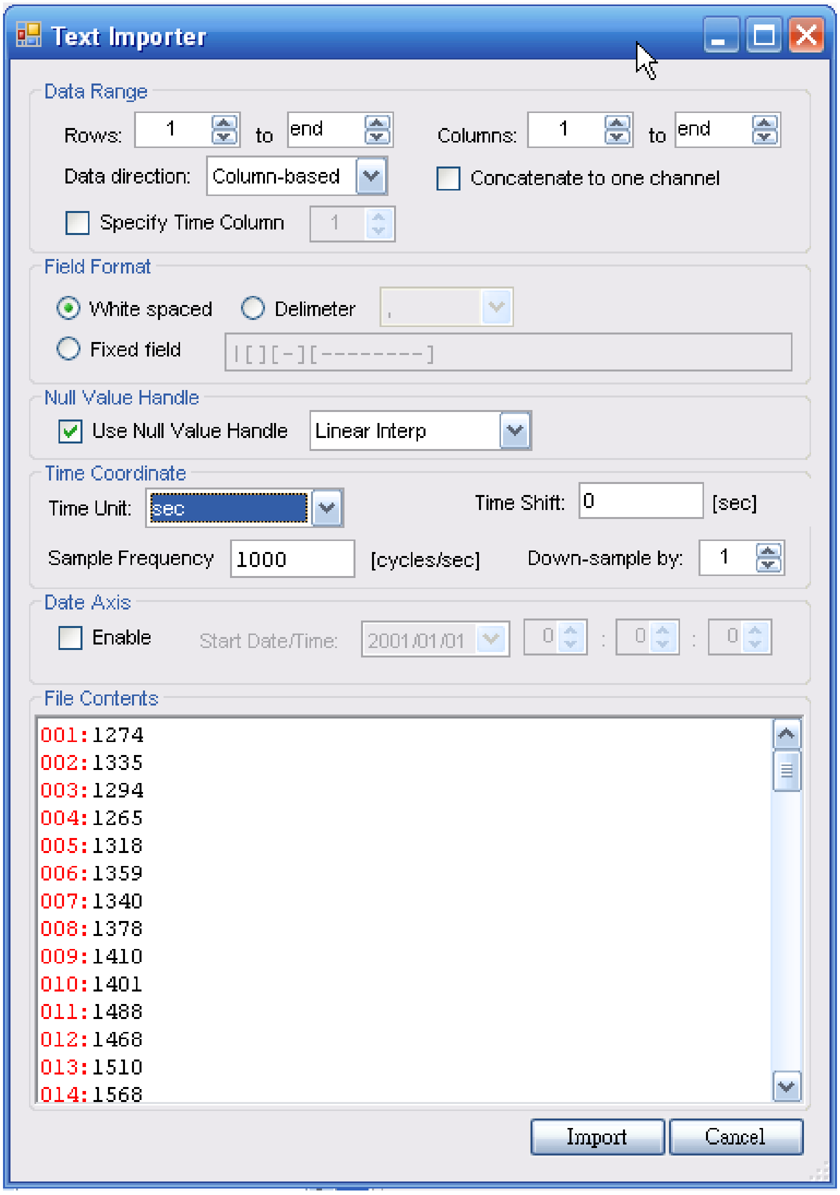


1.3.2 Netowork Window Toolbar

Netowork Window Toolbar has four basic buttons to import and export data.  Open data from file,  Save data to file, *Data Viewer* (this function will be explained in detail in Chapter 1.3.4) and  Export data to Excel.

1. Open data from file

   This function allows DataDemon to import files from formats such as .mat, .sac, .hea, txt, csv etc. When importing text based files, a Text Importer (image below) will appear. Configure the settings to translate the text file into information that can be read and processed by DataDemon.



If the imported file is not supported by DataDemon, a warning message will pop up (image below) asking the user whether to read the file as plain text or use Text Importer to import the file.

The user will be required to configure the Text Importer on how a text file is to be read, e.g. how to set the time, frequency, and data range.

File Extension	Description
mat	MATLAB file format.
sac	SAC (Seismic Analysis Code) is used for Seismology.
tfa	DataDemon file format.
csv	Comma-separated values.
wav	Wave file format.
mp3	Mpeg1 audio layer3 file format.
hea	A signal file format used in biomedicine. Pleas refer to www.physionet.org for more information.

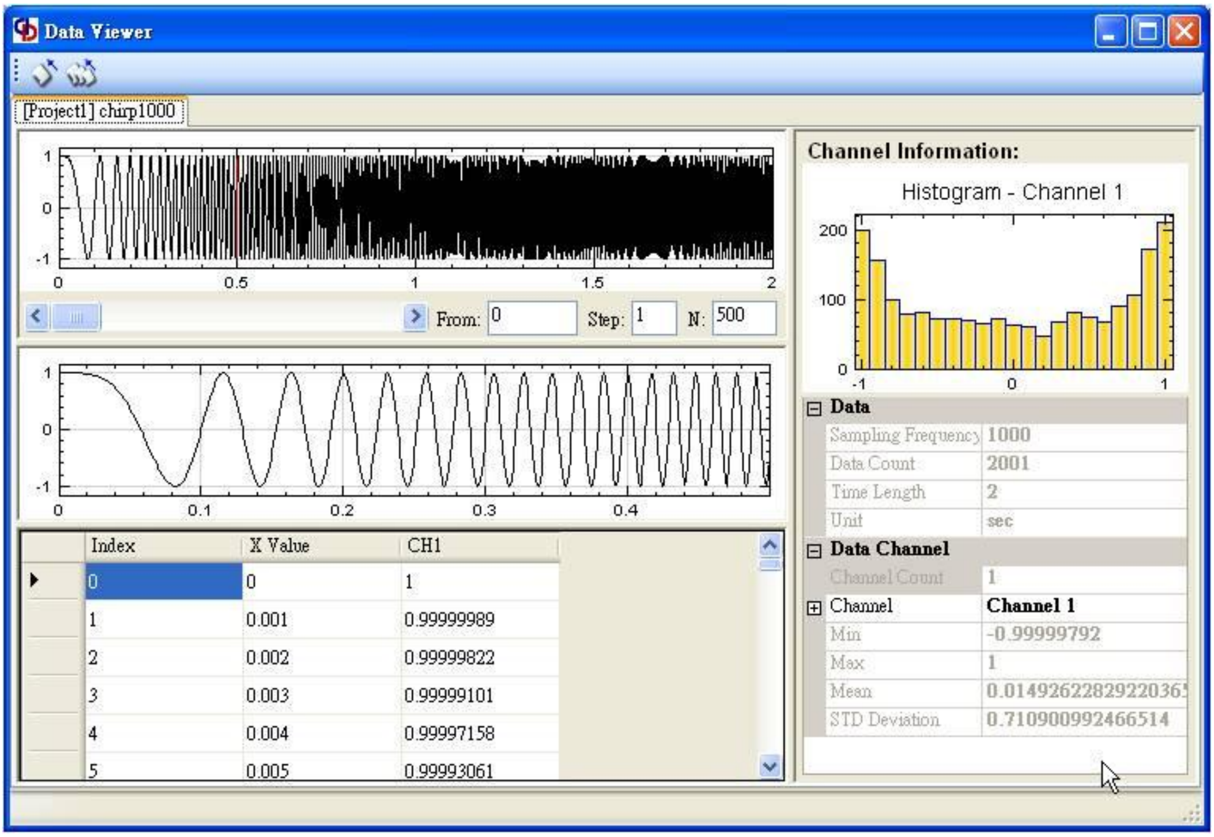
2. Save Data to File, Export data to Excel

These two operations can also be found under Writer Signal Flow Object (SFO). The result obtained from a SFO can be exported and saved into various file formats, such as mat, tfa, txt, csv etc. And sound signals can be saved as wav files.

3. Data Viewer:

Select a SFO (Signal Flow Object) and click on the *Data Viewer* button to

open up a new window detailing the information of the signal.



4. Force Update:

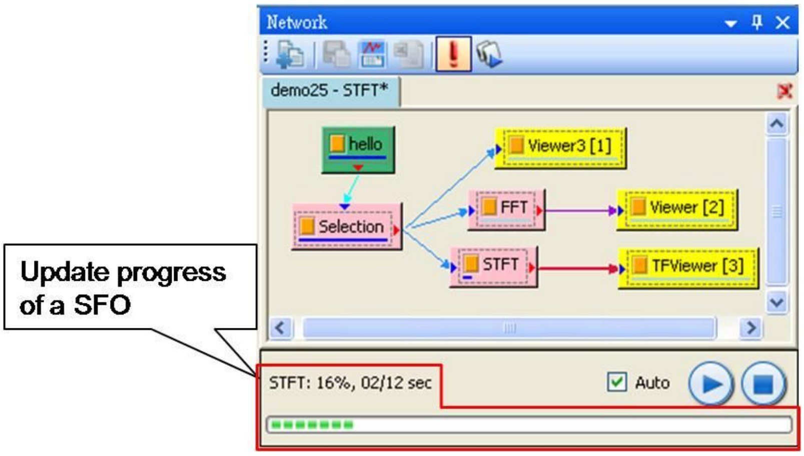
Force Update will force all SFOs to be recalculated again.

5. *Batch Run:*

Batch Run allows user to configure and run multiple instances of the Signal Flow Diagrams created in the *Network Workspace*. Any property variables of the SFO can be edited in the *Batch Run* instance to allow greater flexibility and it saves time for not requiring to edit the original Signal Flow Diagrams in the *Network Workspace*. *Batch Run* allows user to play around with property variables within the Signal Flow Diagrams. And changes made to the variables can be tested through multiple *Batch Run* instance for result comparison.

1.3.3 Network Control Area

At the bottom of the *Network Window* is the *Network Control Area*. *Network Control Area* shows you the update progress of a SFO and the controls to ☑ Auto *Automatically Update Calculation*, *Update Calculation* and *Stop Calculation on a SFO*.



1. Auto:

    You can check the box *Auto* to automatically update any SFO you have edited or added. This can save you time if the SFO doesn't require much time to update. But there will be some SFOs which will require long period of time to compute and it is not efficient to update them automatically every time they are edited. If you do not want to automatically update SFOs, you can un-check the *Auto* box and click on to update a selected SFO manually.

2. Update Calculation:

    When *Auto* is not checked, click on button to update a SFO that has been edited. A SFO which require updating will have a sky blue line near the bottom of the SFO. Once a SFO is updated, the sky blue line will turn into a dark blue line that indicates an updated SFO.

3. Stop Calculation:

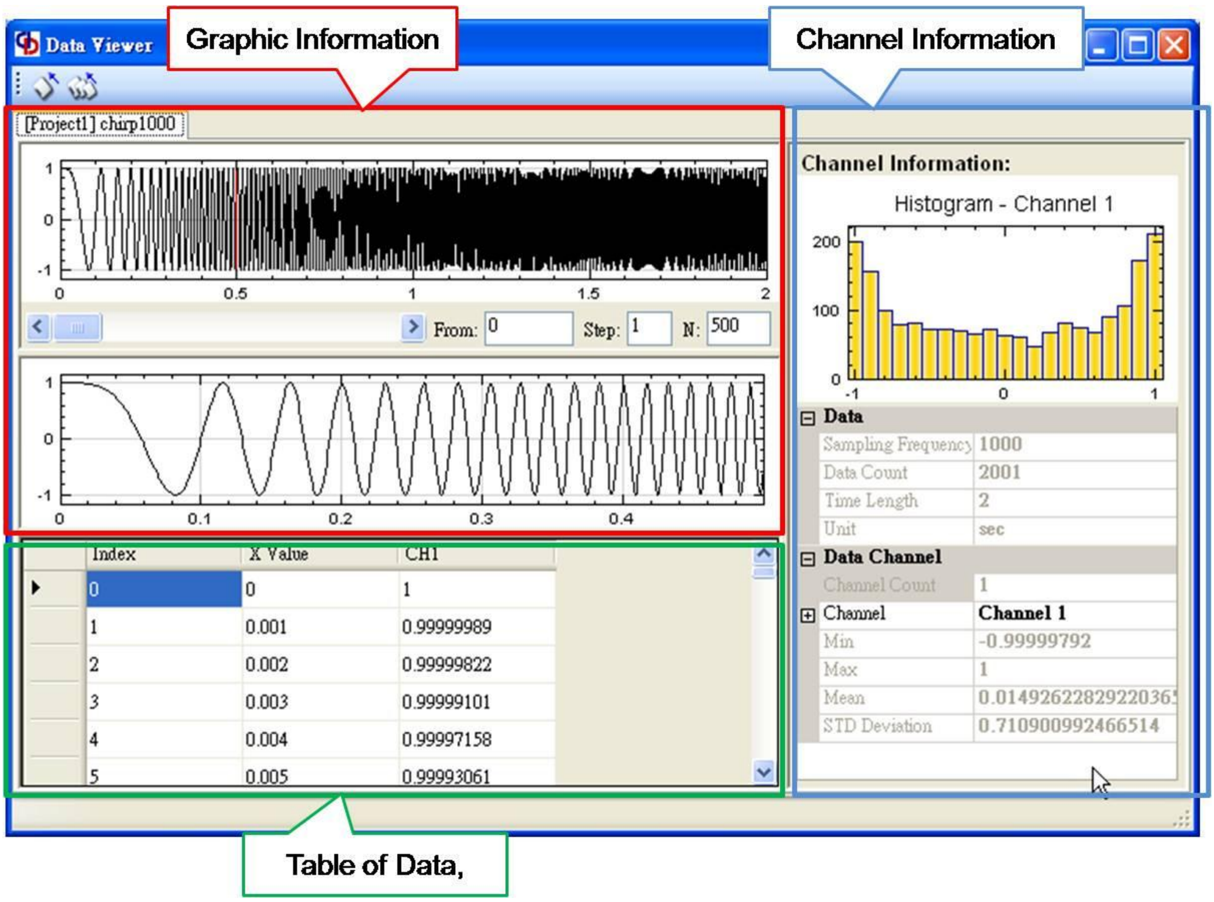
    During an update of a SFO, you can click on button to stop the update.

1.3.4 Data Viewer

Data Viewer can be created so that the calculation result of a SFO can be quickly viewed, examined and analyzed through an information browser.

Select a SFO and click on the *Data Viewer* button and the *Data Viewer* window will pop up. Depending on the type of selected SFO, *Data Viewer* will display data based on the different signal types such as signal source, spectrum, numeric and spectra. The following will be a brief introduction to the interface of the *Data Viewer* and how it displays information depending on the different types of SFO selected.

The interface of the *Data Viewer* is divided into three sections (as shown in the image below); the upper left half is the *Graphic Information*, where the graphs are displayed, the bottom left half is the *Table of Data*, which contains the values of the graph which is displayed in a table format and the right half is where the *Channel Information* is shown (e.g. sample frequency, data count, time length and unit). All three sections will show different information based on the type of SFO selected.



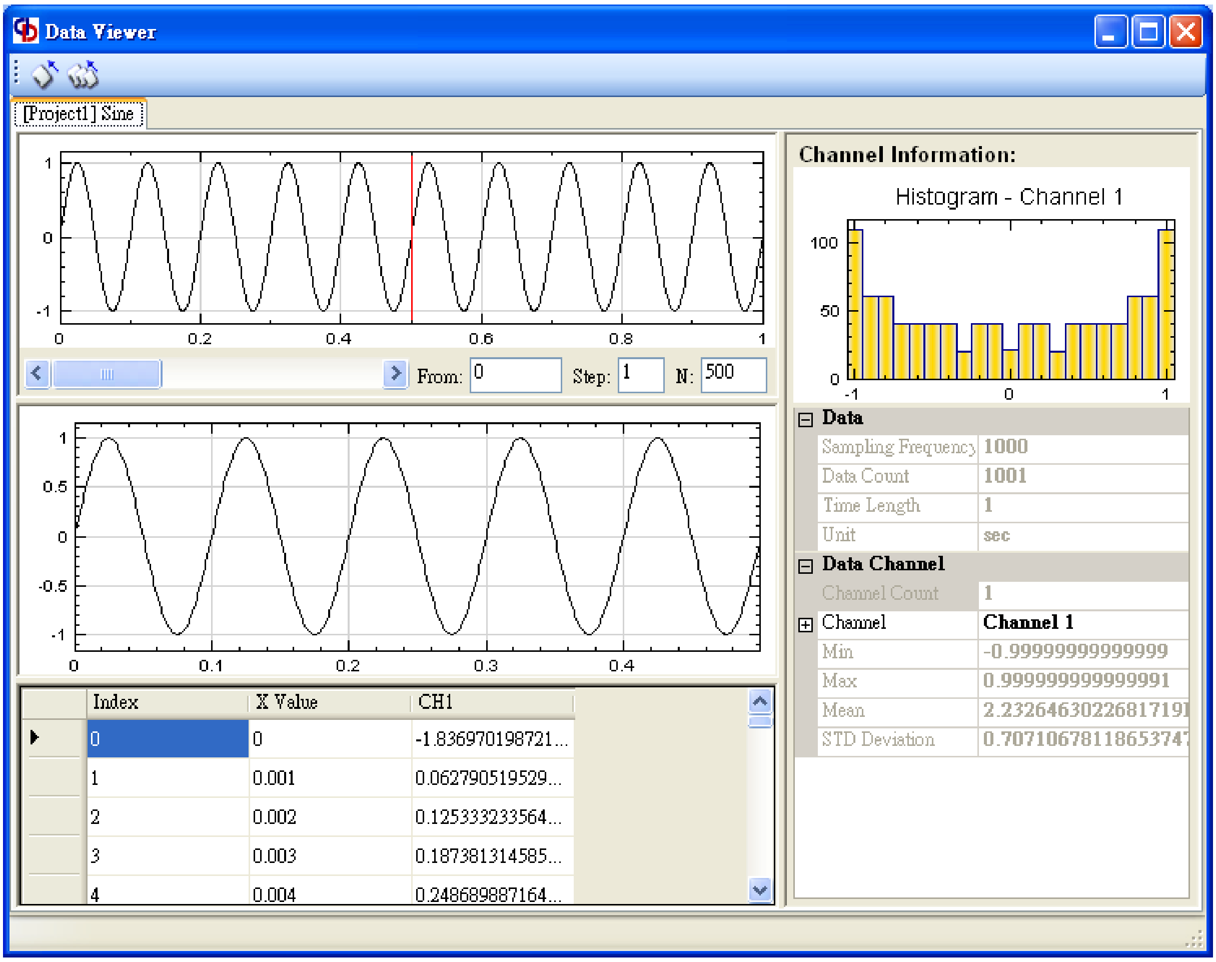
On the top left corner of the *Data Viewer*, there are two buttons; *Close Tab*

and *Close all tabs*. A *Data Viewer* can open many SFOs and the multiple *Data Viewer* profiles will be displayed as tab menus under the *Close Tab* and *Close all tabs* buttons. Click on *Close Tab* to close the current *Data Viewer* profile and click on *Close all tabs* button to close all *Data Viewer* profiles. (**Note:** *Data Viewer*'s content will not change with a new update of a SFO. To view the latest update of a SFO in *Data Viewer*, you must close the *Data Viewer* profile and reopen it again in the *Data Viewer*.)

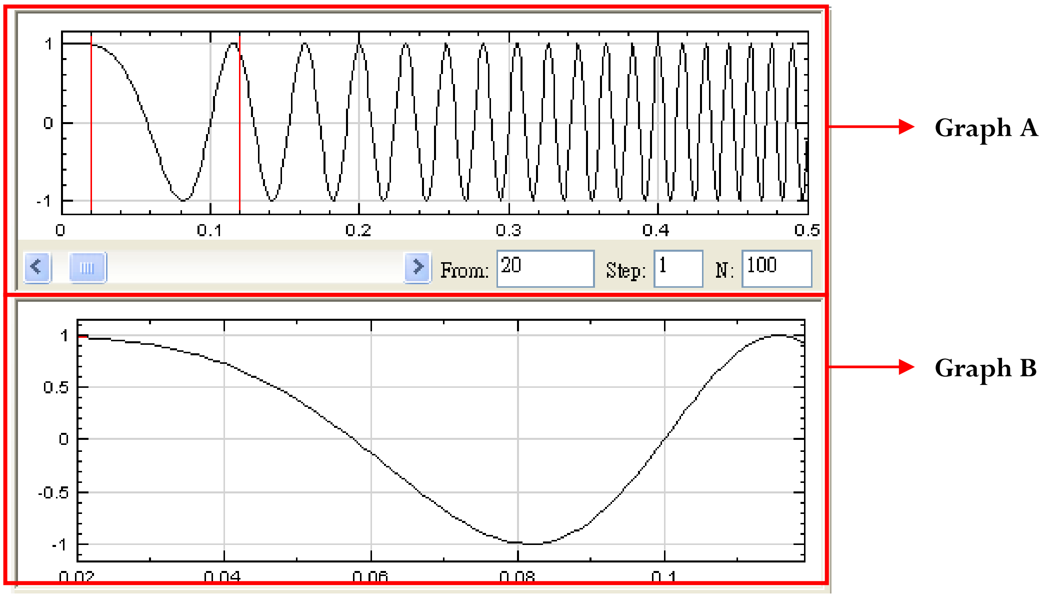
The following will explain the different types of *Data Viewer* interface based on the different types of SFOs.

1. Signal

Graphic Information is where the graphs are drawn and the graphs are drawn based on X-axis (time) and Y-axis. There are two graphs, the top *Graph A* displays the whole graph that is plotted along the X-axis. The bottom graph *Graph B* can be a detailed zoomed-in version of the top graph and the area shown in *Graph* B is the section between the two vertical red lines displayed in Graph A (**Note:** Depending on the configuration, it is possible for the vertical red lines to span across the entire *Graph A*). The field *From* indicates the starting position of the left vertical red line (the starting position begins at From=0), the field *Step* indicates the counting increment that will be shown from *Graph A* to *Graph B* (e.g. having Step=2 will have *Graph B* showing every second value of the points in *Graph A*). The field *N* indicates the sampling area of *Graph B* (**Note:** The maximum value of N allowed is relative to the value of Step and the Sampling Frequency of the SFO).



In the image above, Graph B's first point begins at 0, contains 500 points and has not been down sampled.



You can move the horizontal scroll bar which is located below *Graph A* to reposition the two vertical red lines in *Graph A*.

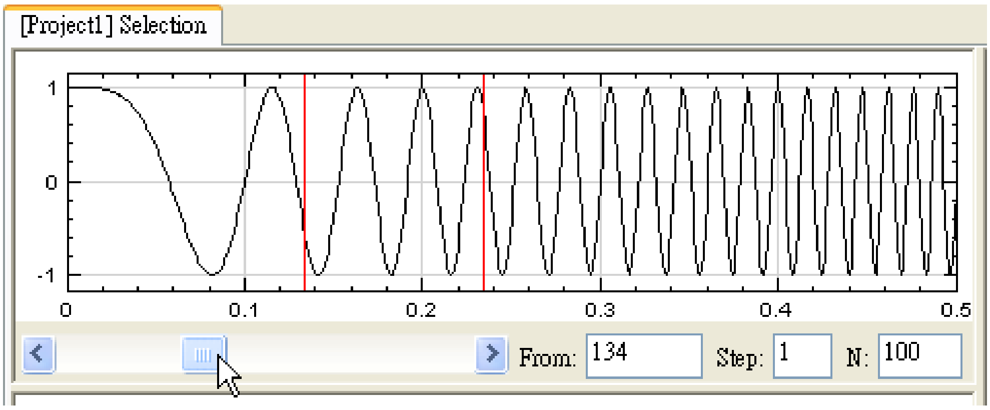


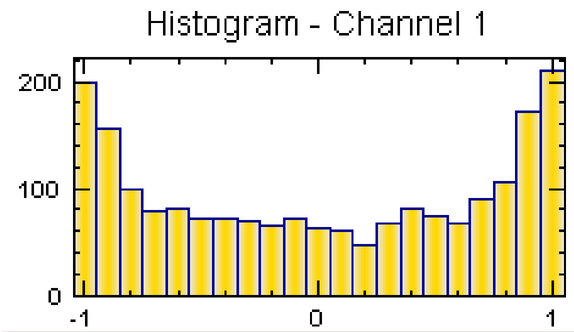
Table of Data is located at the bottom left half of the *Data Viewer* window. The table is column-based, each row represents a different point in time. The first column is the Index (a number indicating the point on the graph), the second column is the X Value and followed by the different channels, CH1 (Channel 1), CH2, CH3 etc. In the image below there is only CH1 available.

Index	X Value	CH1
0	0	0
1	0.001	6.283185303045...
2	0.002	0.000251327409...
3	0.003	0.000565486647...

The values in the Table of Data are dependent on the configuration of *Graph B* in

Graphic Information. As the position of *Graph B* is moved, the changes will also be updated in the Table of Data.

Channel Information is located on the right hand half of the *Data Viewer* window. Channel Information displays the statistical information of the SFO. The Histogram is a bar chart representing a probability distribution and the information is divided into 21 bars.

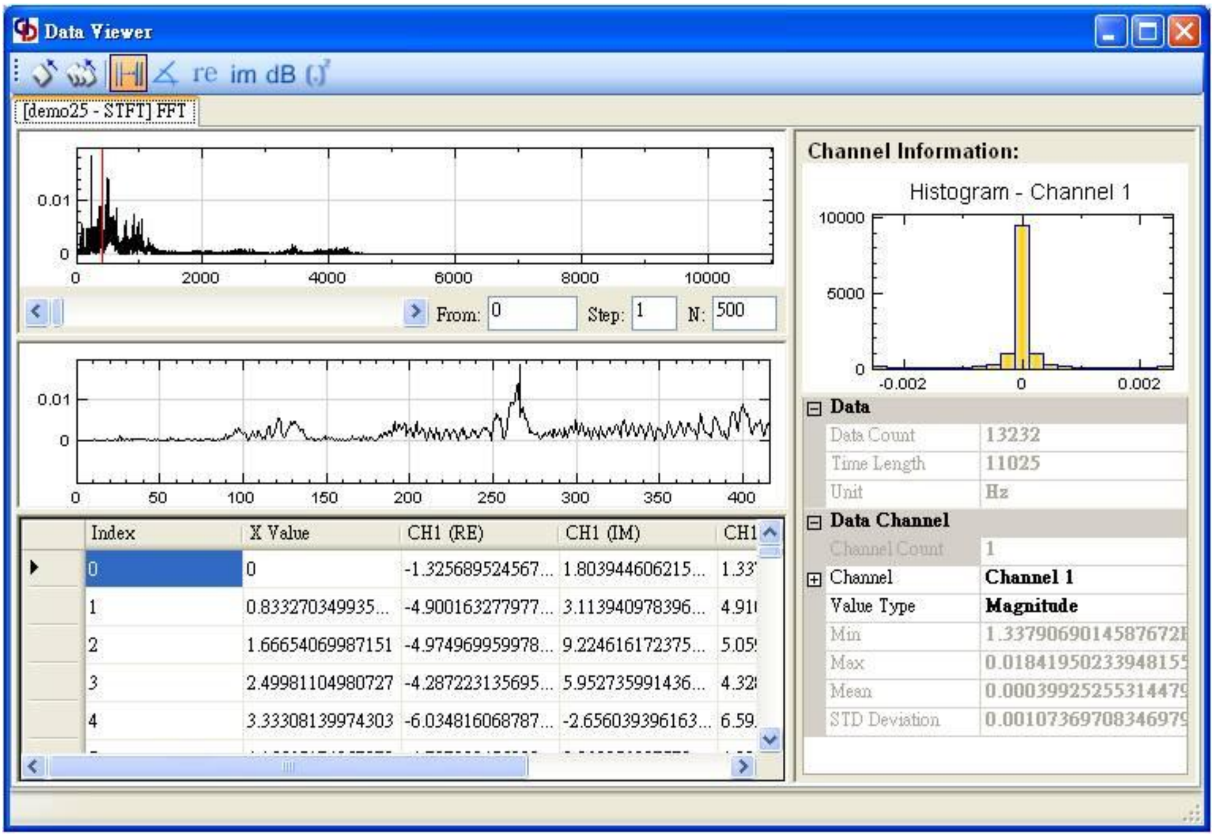


Data	
Sampling Frequency	1000
Data Count	2001
Time Length	2
Unit	sec
Data Channel	
Channel Count	1
Channel	Channel 1
Min	-0.99999792
Max	1
Mean	0.0149262282922036
STD Deviation	0.710900992466514

Below the Histogram graph, there is the Data and Data Channel information. The information shown here cannot be edited. (**TIPS:** If you wish to edit this information, you will have to go back to the *Network Workspace* to select the SFO and edit the values in Properties). In the Channel field, you can select the Channel to be displayed through the drop-down menu.

2. Spectrum

Click on a Spectrum source SFO and click on the *Data Viewer* button will open a *Data Viewer* with 6 additional buttons , , , re, im, dB, (.)^2 (magnitude, phase, real part, imagine part, gain and power spectrum buttons) at the top left corner of the *Data Viewer*. Each button will change the looks of the graphs and the channel data based on the option selected.

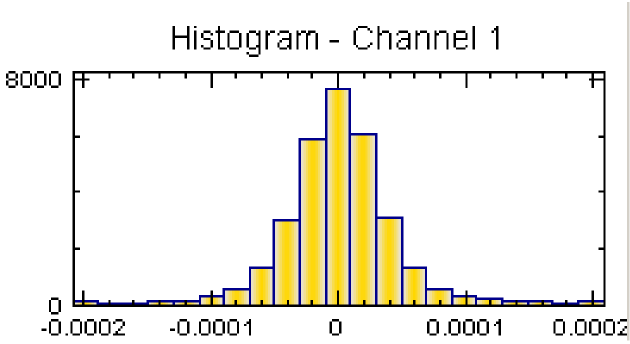


Bottom left screen contains the table which holds the information of the spectrum. X Value represents frequency and each channel contains four column vector: Real part (RE), Imaginary part (IM), Magnitude (MAG), Phase (PHASE).

Index	X Value	CH1 (RE)	CH1 (IM)	CH1 (MAG)	CH1 (PHASE)
0	0	2.455269054735...	0	2.455269054735...	0
1	10.0003200102403	-6.737160609830...	-0.000551901395...	0.000555998276...	-96.95976583867...
2	20.0006400204807	0.000107569773...	-0.001446670680...	0.001450664438...	-85.74749606478...
3	30.000960030721	-5.679091216591...	0.001241813580...	0.001243111490...	92.618439712872
4	40.0012800409613	0.000194001695...	0.000319203871...	0.000373534160...	58.71013132168...
5	50.0016000512016	-0.000360690876...	0.000542531736...	0.000651489519...	123.6171155060...

Histogram on the top left corner displays the spectrum's probability distribution.

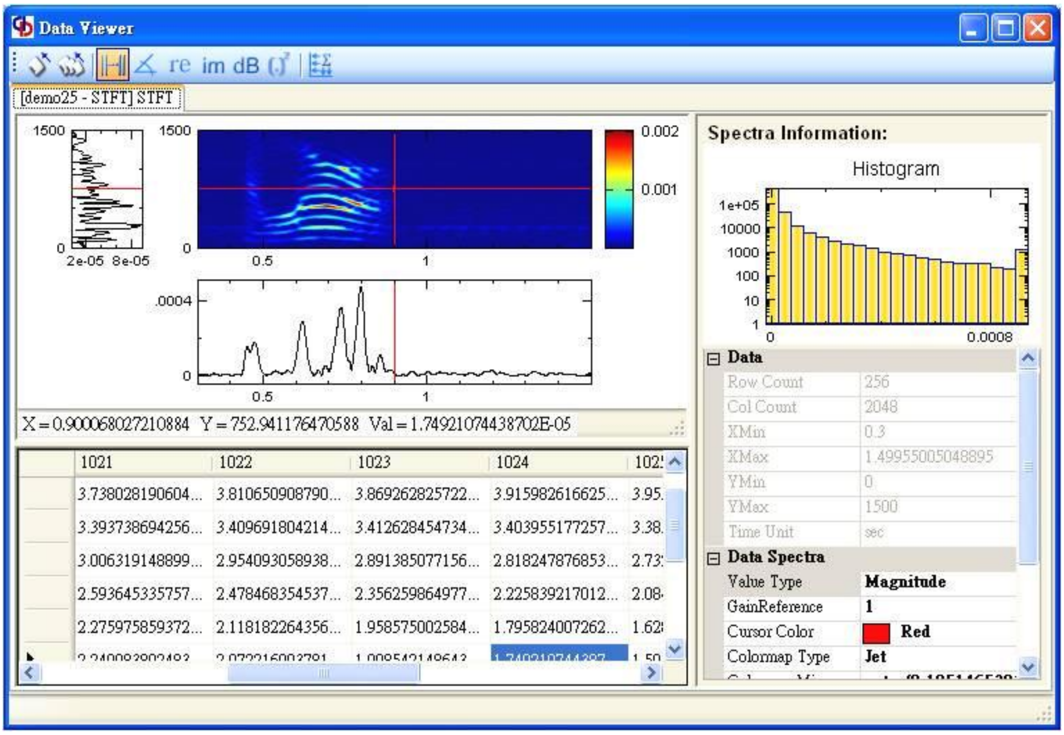
To view different channel and different value type in the histogram, edit the *Properties/Data Channel → Channel* to change the channel and edit the *Properties/Data Channel→Value* Type to change the value type to view.



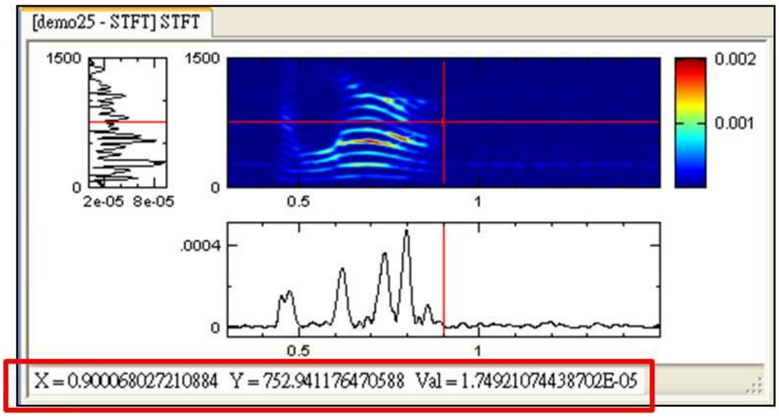
⊟ Data	
Data Count	31250
Time Length	312500
Unit	Hz
⊟ Data Channel	
Channel Count	1
⊞ Channel	**Channel 1**
Value Type	**ImagPart**
Min	-0.0030456070388992021
Max	0.0052127150711840471
Mean	-2.1905842525464665E-07
STD Deviation	6.6143820857034655E-05

3. Spectra

Another type of *Data Viewer* is spectra. Spectra is created through wavelet transformation or other Time-Frequency Analysis and it is a two dimensional complex array.



The graph displayed by Spectra: X-axis displays the time and the y-axis displays the frequency. Depending on the settings (e.g. Magnitude, Phase etc), each point on the graph will change its color based on the new calculated values.



The red cross on the graph displays the value of the current point in the lower left corner. The point can be changed by moving the mouse over to the chosen spot and left click the mouse to select the desitination.

The graph on the left of the main graph displays all the values along the vertical red line marked on the main graph. On the left graph, the y-axis is the frequency and the x-axis is the index value. The graph on the bottom of the main graph displays all the values along the horizontal red line marked on the main graph. On the bottom graph, the y-axis is the index value and the x-axis is the time. Click on the Marginal Freq Time to switch the left graph and the bottom graph to time-frequency analysis.

	387	388	389	390	391
	0.364637702309...	0.366564315190...	0.368525564970...	0.370520297677...	0.3725473
	0.391597575038...	0.393183842198...	0.39480019953384	0.396445757207...	0.3981196
	0.416771718191...	0.418058873476...	0.419371336655...	0.420708422137...	0.4220694
	0.439524551414...	0.440549033414...	0.441594115404...	0.442659267827...	0.4437439
	0.459315403474...	0.460109571242...	0.460919881930...	0.461745925032...	0.4625872
▸	0.475697641935...	0.476290574578...	0.476895538827...	0.477512215506...	0.4781402

The image above shows the information of the Spectra, the information is stored in the maxtrix format. The default display is Magnitude, the time, and frequency values of the table that are represented in the same way as the graph.

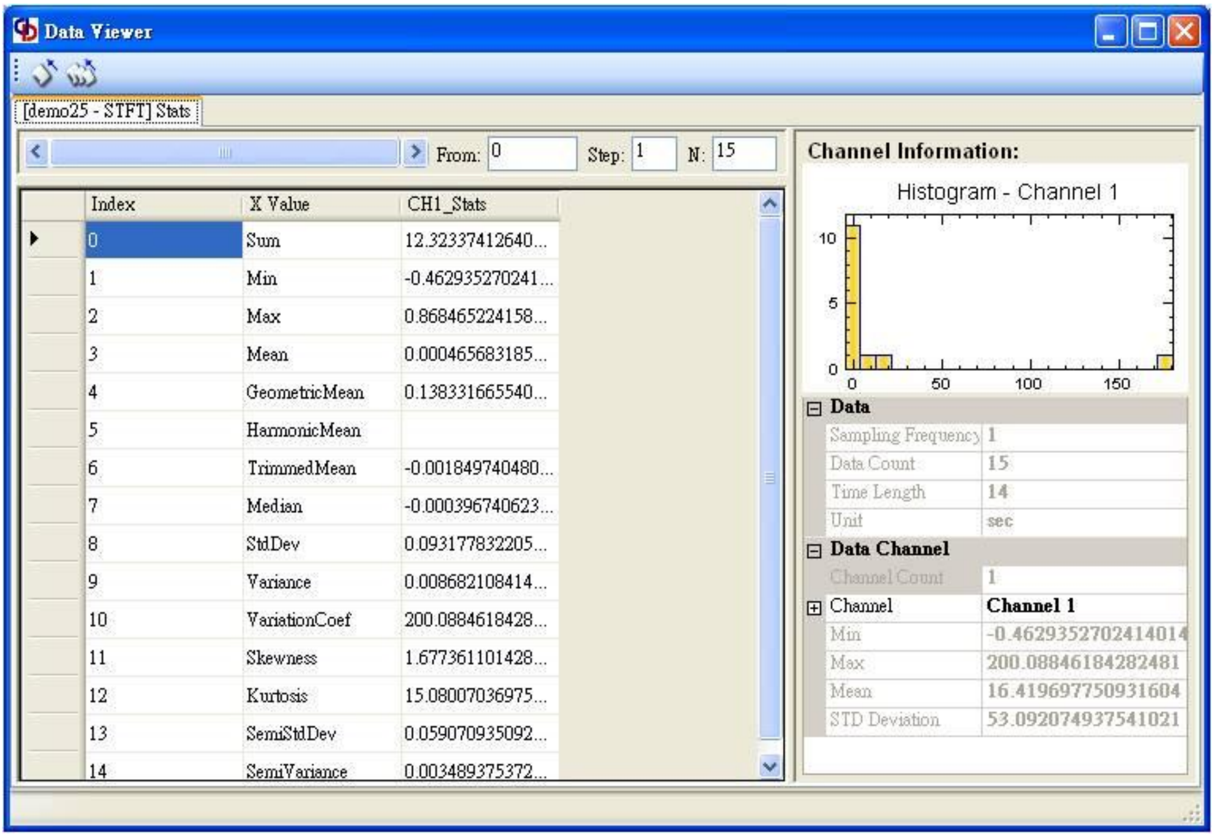
In the *Properties Window* on the right bottom corner, you can edit how the histogram is to be displayed e.g. Colormap Type, ColorMap Min, Colormap Max etc.

YMin	0
YMax	25
Time Unit	sec
Data Spectra	
Value Type	**Magnitude**
GainReference	**1**
Cursor Color	**Red**
Colormap Type	**Jet**
Colormap Min	**auto (0)**
Colormap Max	**auto (0.98170954235222**
Value Min	0
Value Max	0.98170954235222774
Value Mean	0.26125165896214286
Value StdDev	0.29237557416129245

4. Numeric

Numeric data type does not output a graph disaply (e.g. Basic Statistics, Quantiles and Quartiles, Orthogonality Matrix etc). This data type will only be displayed in the table format. Below is an example of a Basic Statistics SFO opened

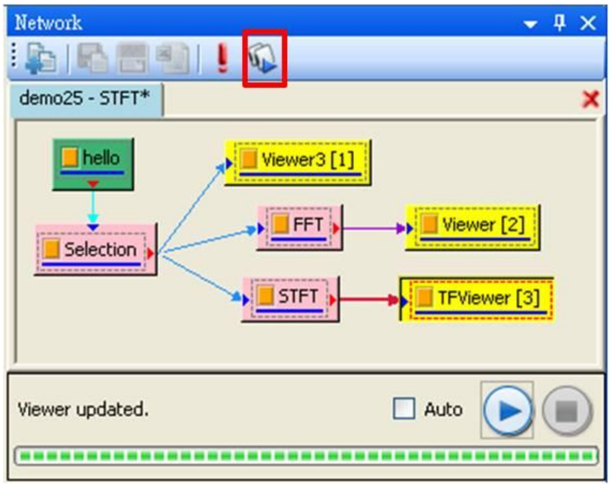
in *Data Viewer*.



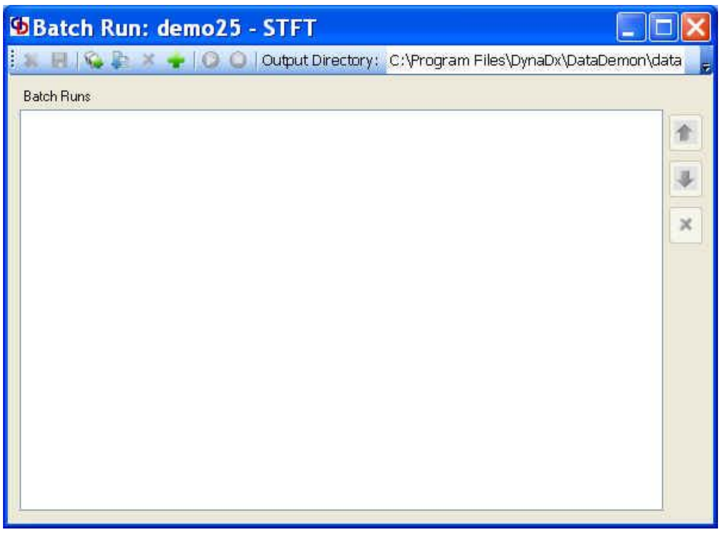
1.3.5 Batch Run (Professional Only)

Batch Run allows the user to simulate one or more *Batch Run* calculations based on the current Signal Flow Diagrams in the Network Windows. The usefulness of this feature is that the user can edit the properties variables of any SFOs in the *Batch Run* and create multiple runs to easily test out the different calculation results. The edited properties variables within *Batch Run* do not alter the original settings of the Network Window's SFO network relationship. So the user can safely edit any properties of the SFOs in *Batch Run* without worrying about influencing the original status of the project.

From the *DataDemon Toolbar* click on Open Project button (or type Ctrl+O) and open the file demo 25 located in C:\Program Files\DanaDx\DataDemon\Data.



After clicking Open Batch Run Dialog button as shown in the above figure. The *Batch Run* window interface is shown below.

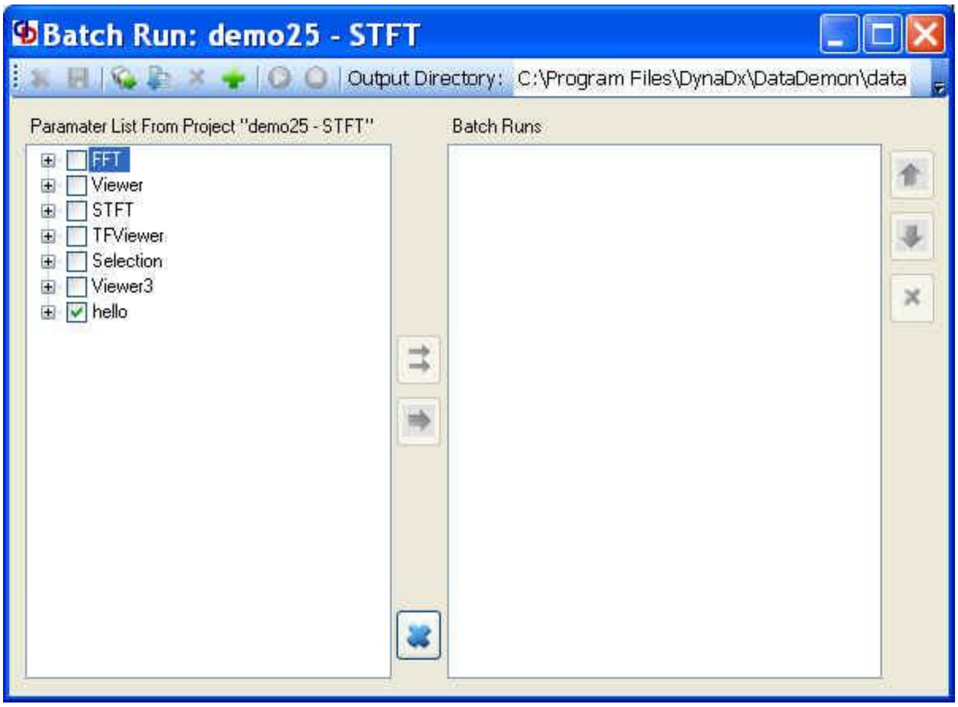


The fuction for each button is,

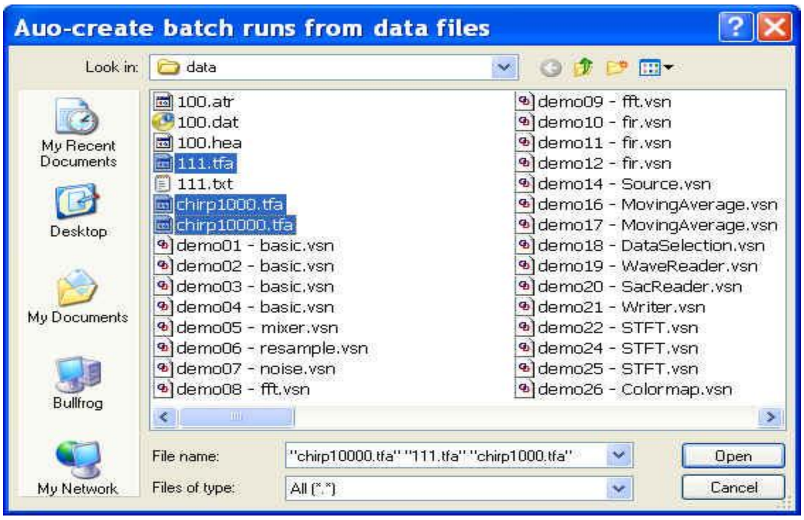
1. Click on to clear all *Batch Runs* and click on to save *Batch Run* to the project.

2. Click on to delete the selected *Batch Run* item from the list.

3. Click on to open the *Parameter List* which contains modifiable components to be applied to selected runs. If you have a component that you wish to add to all the *Batch Run* profiles then you can click on to add it to all *Batch Run* profiles. Or else use to add one component to the selected *Batch Run* profile. Click on to hide the *Parameter List*.



4. Both and allow the user to create *Batch Run* profiles. allows the user to create and edit *Batch Runs* from the exisiting SFO parameters. The user can pick and choose the parameters to alter for each *Batch Run*. can only be clicked when a file has been imported into the project and the DataSource is checked from the Parameter List. (**Note:** DataSource parameter will only be available in the *Batch Run* window after a file has been imported into the project). Clicking on the will open a browser window to select files to be added to the *Batch Run*. (**TIPS:** For multiple files selection, hold the Ctrl key)



5. Click on to execute a *Batch Run* and click on to stop a *Batch Run*.

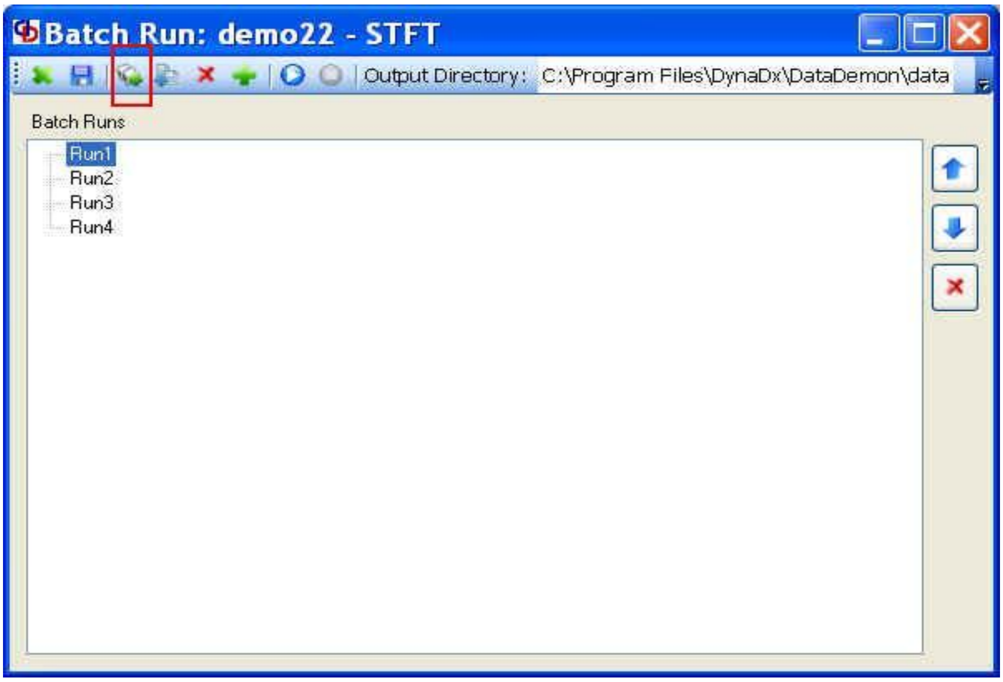
6. *Output Directory* is where the result of a *Batch Run* will be saved. If the source of the *Batch Run* is already a saved file, then the *Output Directory* will be automatically named to match the saved source file name (e.g. Source file name is C:\Project.vsn then the Output Directory will be set as C:\Project). If the source file is newly created and haven't been saved yet, then the *Output Directory* will have to be manually entered in the *Output Directory* field or click on to designate an output directory from existing folders.

After a *Batch Run* has been successfully executed, all the **Viewers** included in the *Batch Run* will automatically output a graph and if **Writer** is connected then **Writer** will output signals into the *Output directory*.

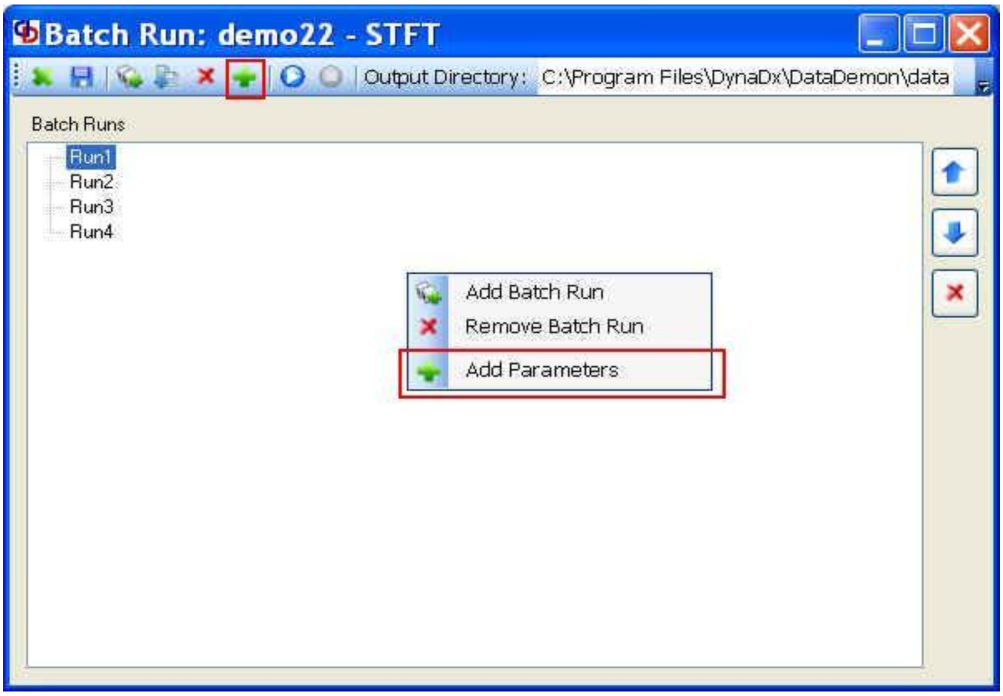
Shown below are two examples of the *Batch Run* process:

- Example 1: To modify the parameter of a *Batch Run*.

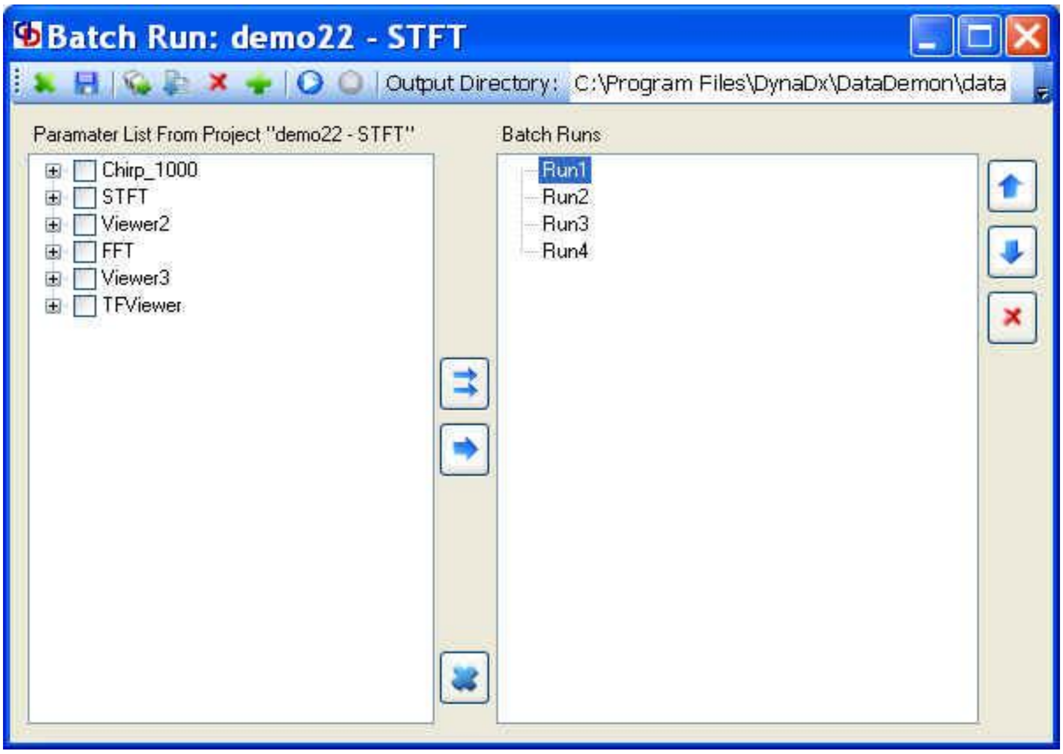
After open (Crtl+O) project demo 22 and open the batch run dialog, click on button four times to create four Run profiles (*Batch Run* profiles). Currently the new Run profiles are empty.



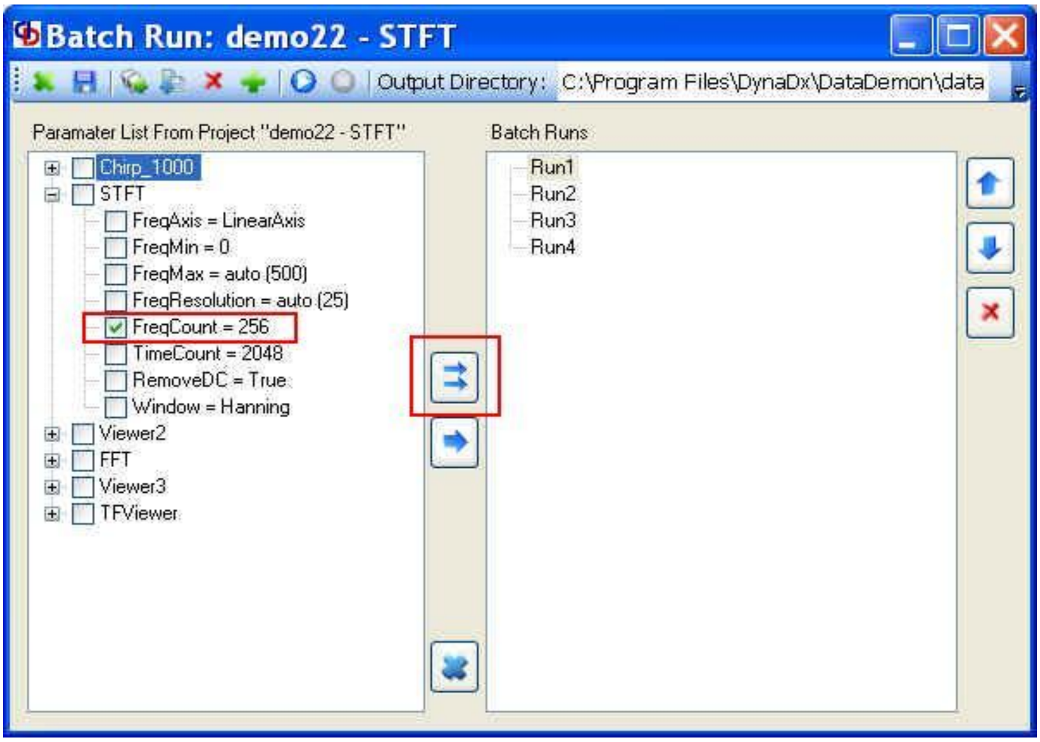
Click on Add Parameters to open the *Parameter List*. You can also add parameters through right clicking on the *Batch Run*s window from the mouse menu (shown in the image below).



After clicking on *Add Parameters* from the mouse menu, the *Parameter List* will appear from the left of the *Batch Run* window (as shown in the image below).

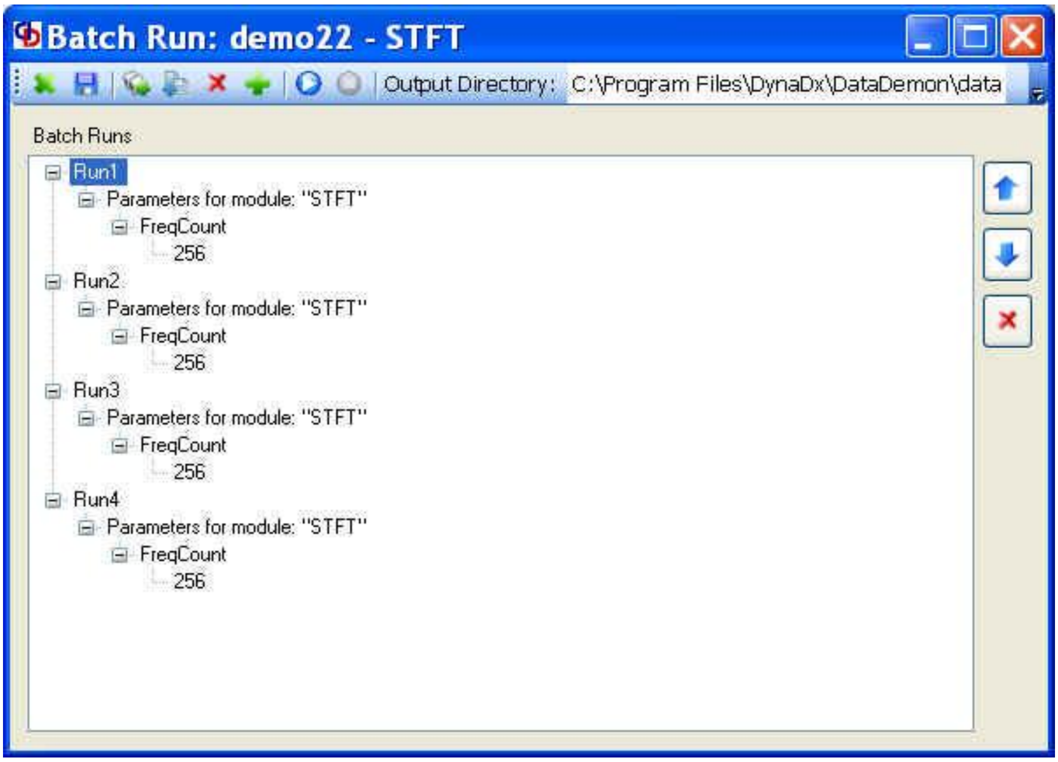


Suppose that you want to change the *FreqCount* value of STFT. First locate the variable *FreqCount* under *Short Term Fourier* (Log-STFT) by clicking on [+] to expand the list. Check the box corresponding to *FreqCount* and highlight is, then click on or to insert the variable to one or more Run profiles.

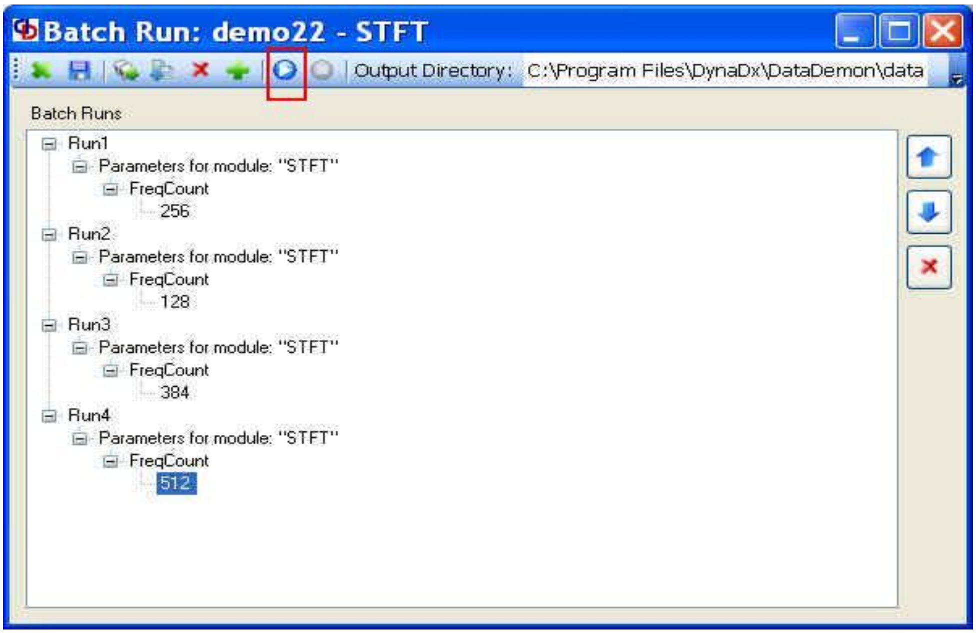


In the example was clicked to be added to all four Run profiles. After adding

FreqCount, click on ✖ button to hide the *Parameter List* and click on all the [+] icon of every Run profiles to expand the details. Now you should be able to see *FreqCount* value as 128 after clicking on all of the [+] icons (as shown in the image below).



It is extremely easy to edit the value of *FreqCount* in any of the Run profiles. Just double click on the value 128 of *FreqCount* to enter the new value (**TIPS:** You will need to double click the mouse on the value slowly because double clicking too fast won't allow you to edit the value).

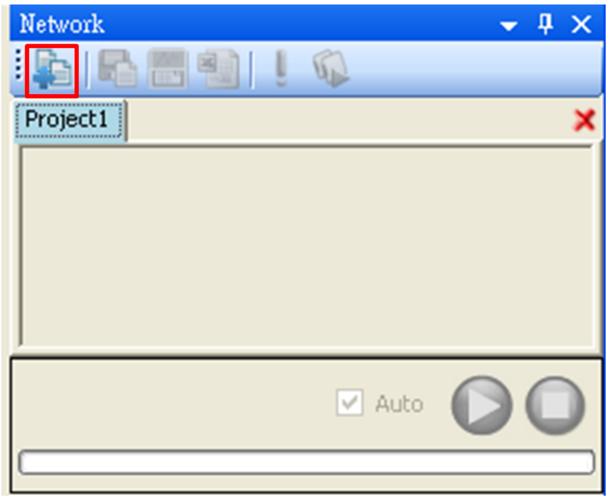


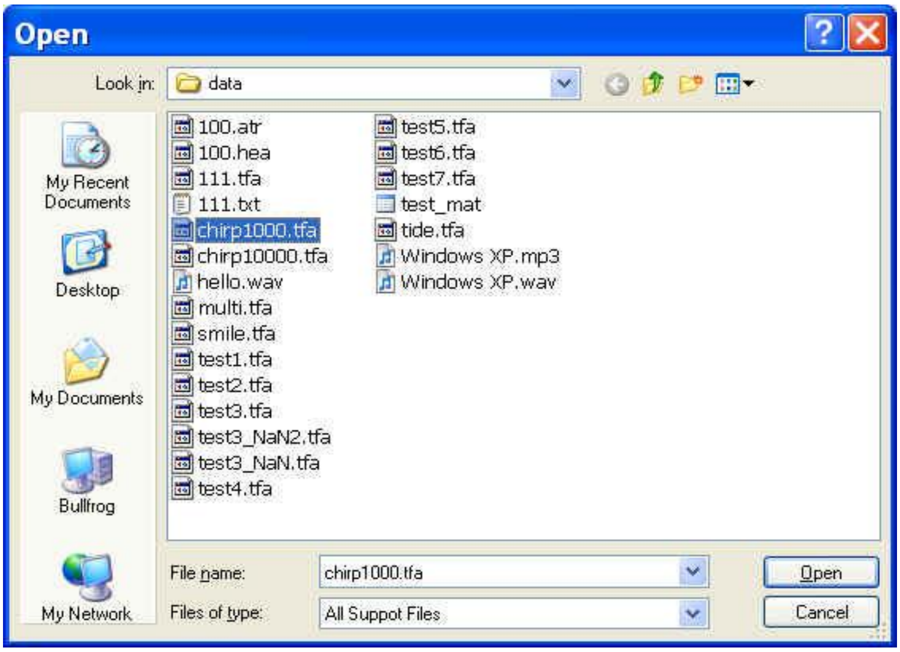
Click on to start the *Batch Run* process. If there is a need to stop the *Batch Run* process, click on the Stop *Batch Run* button. After the *Batch Run*, all results will be shown in the Viewer and the images will be saved in the Output Directory. Each Run profile will generate a single image file and the image file format can be configured in the Preference button of DataDemon.

- Example 2:

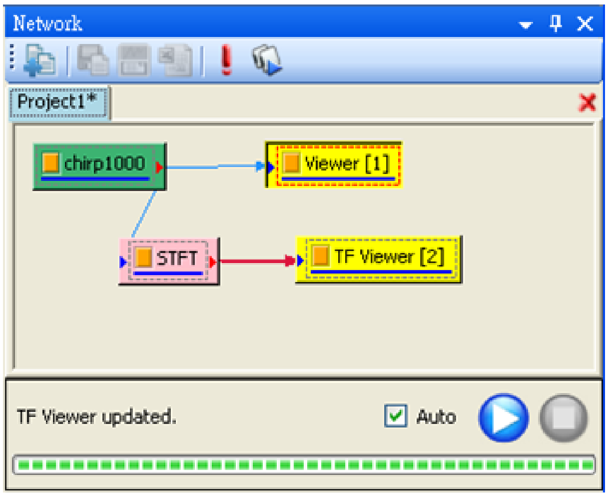
In this example a single .VSN file can be used to calculate multiple data files. Firstly a Signal Flow Diagram will be created and then a demonstration of how to calculate multiple data files will be explained.

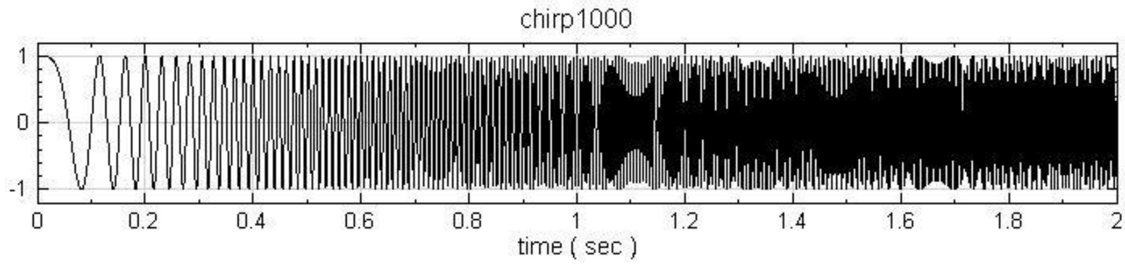
Click on the Import data from file in the *Netowork Window Toolbar* and select Chirp1000.tfa from the location C:\Program Files\DynaDx\DataDemon\data.

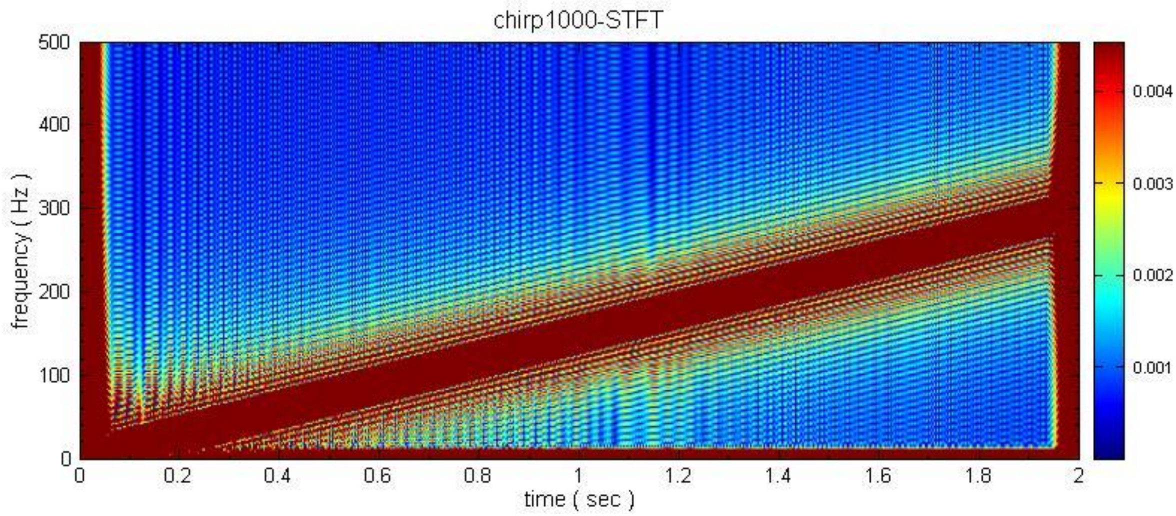




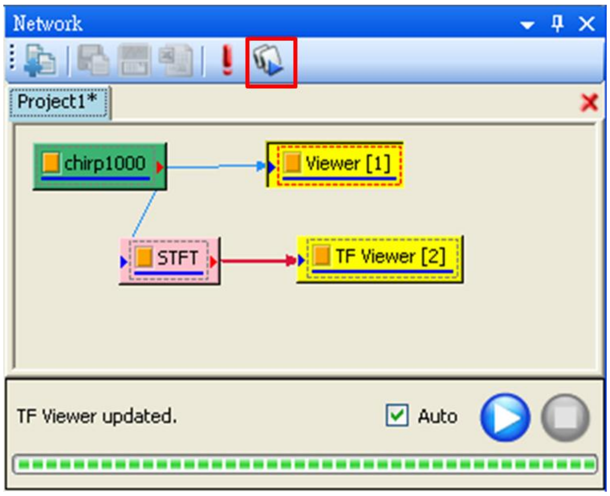
1. After adding the file, add a *Viewer→Channel Viewer* to draw the graph and then use *Compute→TFA→Short Term Fourier Transform* to open another *Viewer→Time-Frequency Viewer* (as shown in the image below).



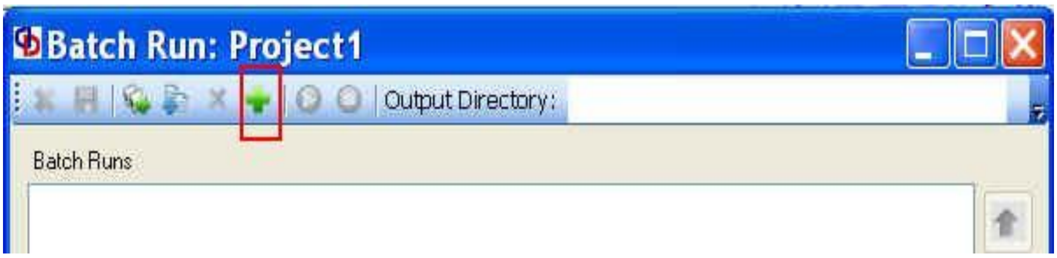




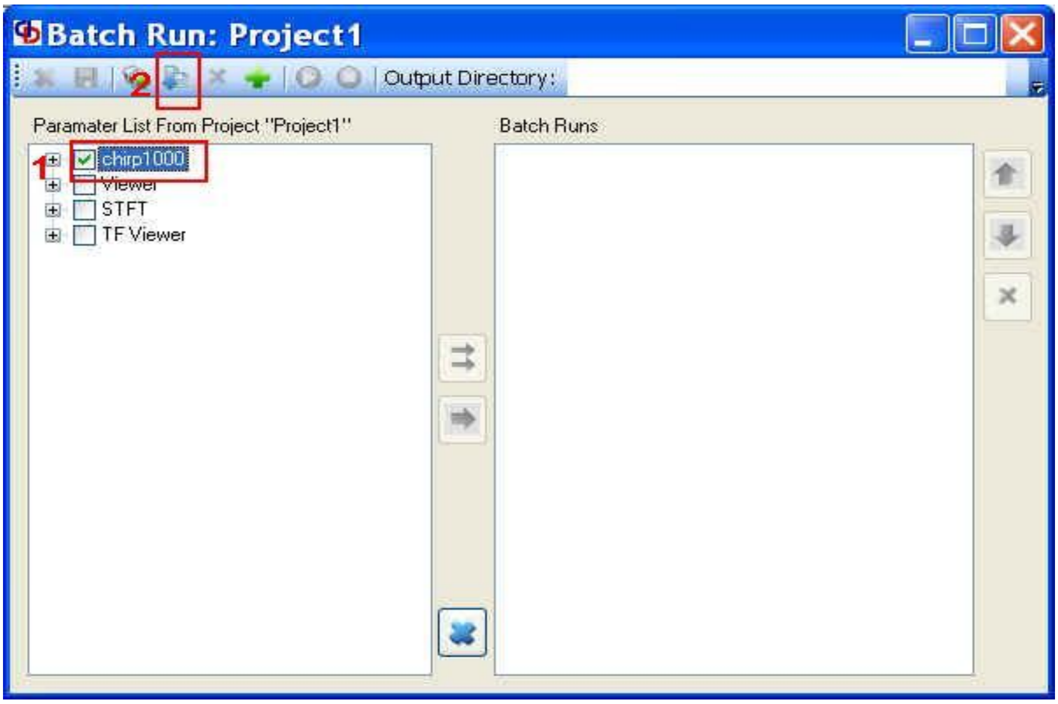
After the Signal Flow Diagram has been created, click on *Batch Run* button to start editing the *Batch Run*.



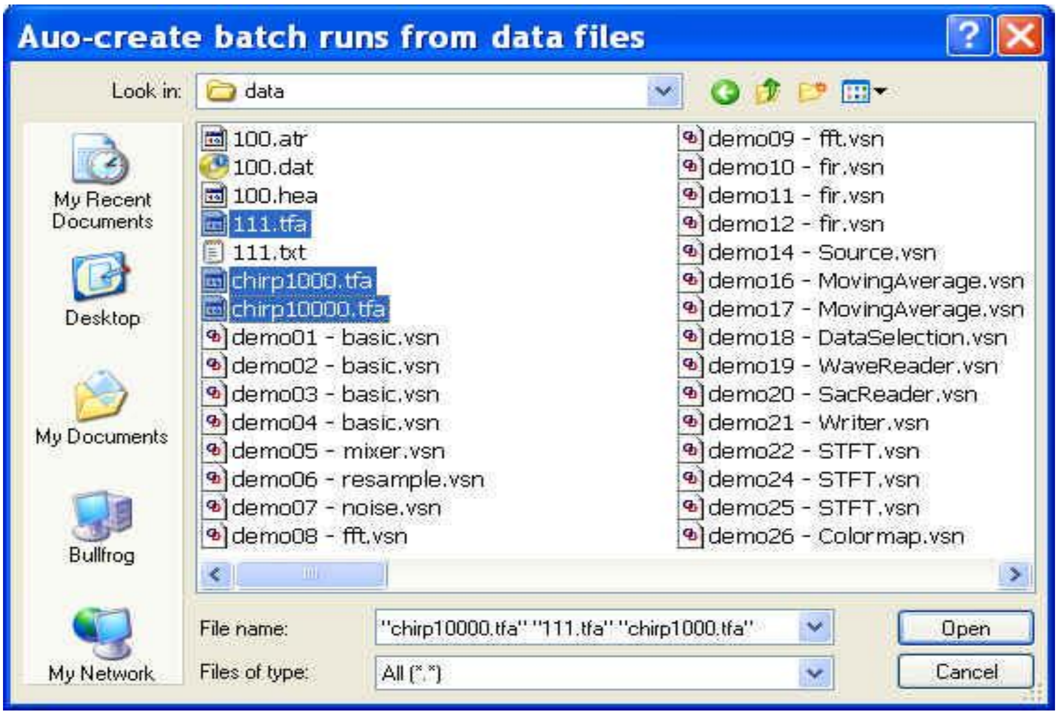
Add parameter: once the *Batch Run* window is open, click on to add parameters.

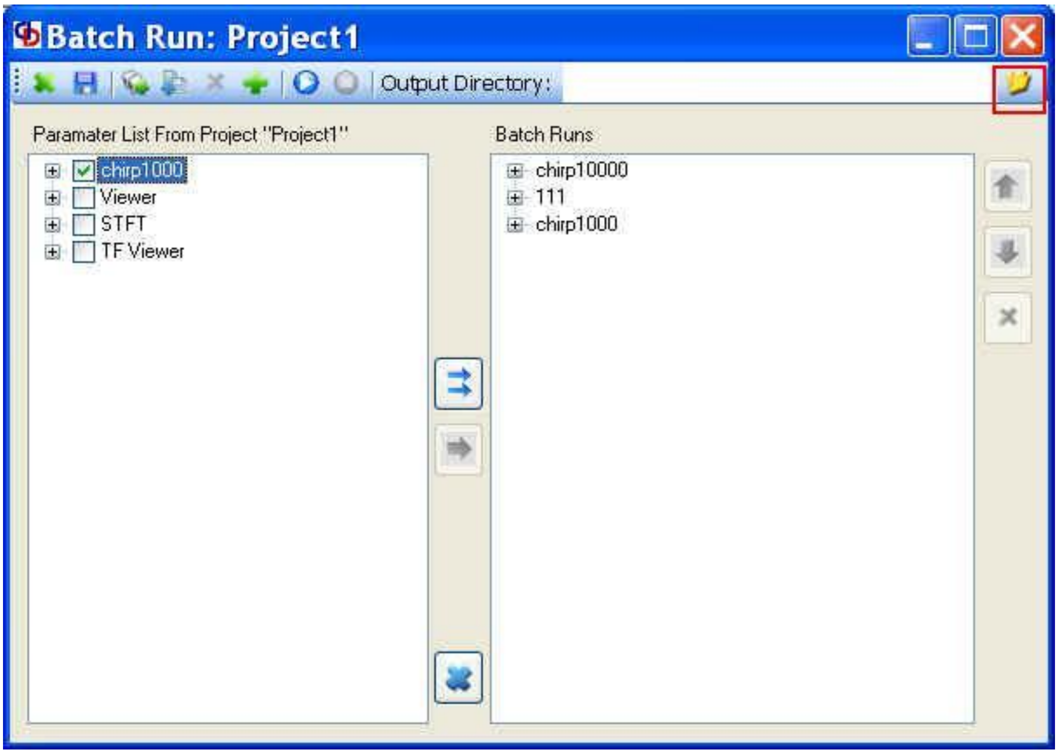


Check the box next to DataSource (chirp1000) in the Parameter List. If there are multiple signal files, then you will need to expand [+] DataSource and edit the files you want. Next click on to select multiple data files.



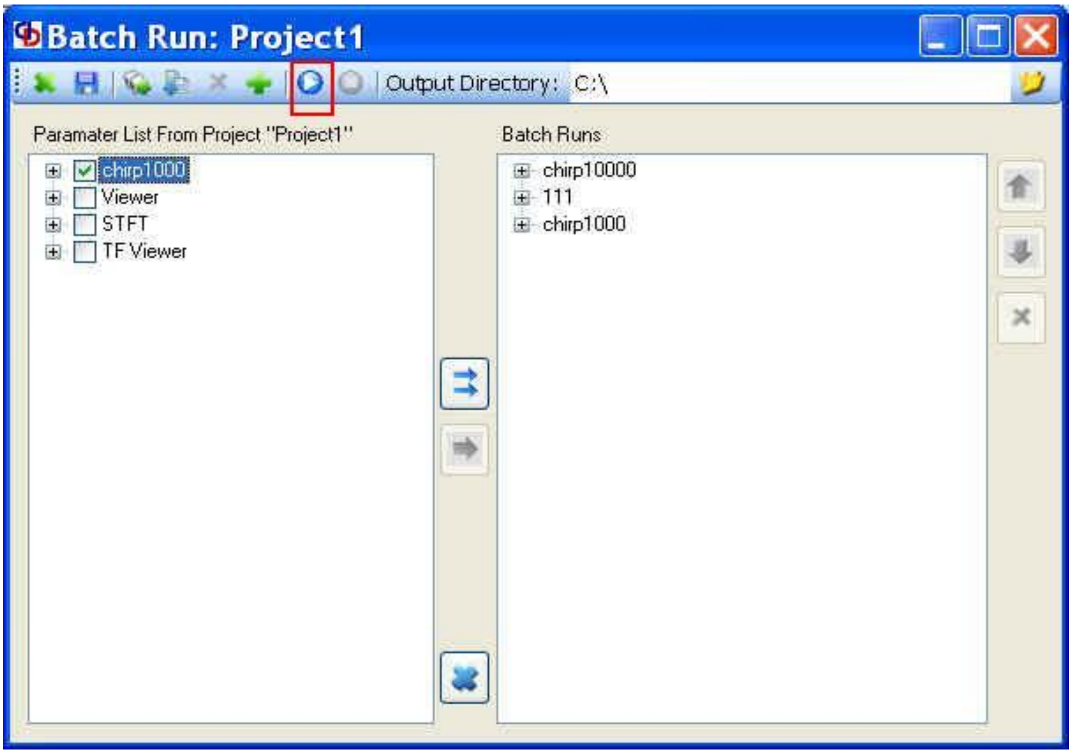
Choose 111.tfa, chirp1000.tfa, and chirp10000.tfa three files which are located in C:\Program Files\DynaDx\DataDemon\data to be used for this *Batch Run*. Now three new Runs will appear on the right-hand side section of *Batch Runs*. If more files are to be added, then click on and repeat the steps.



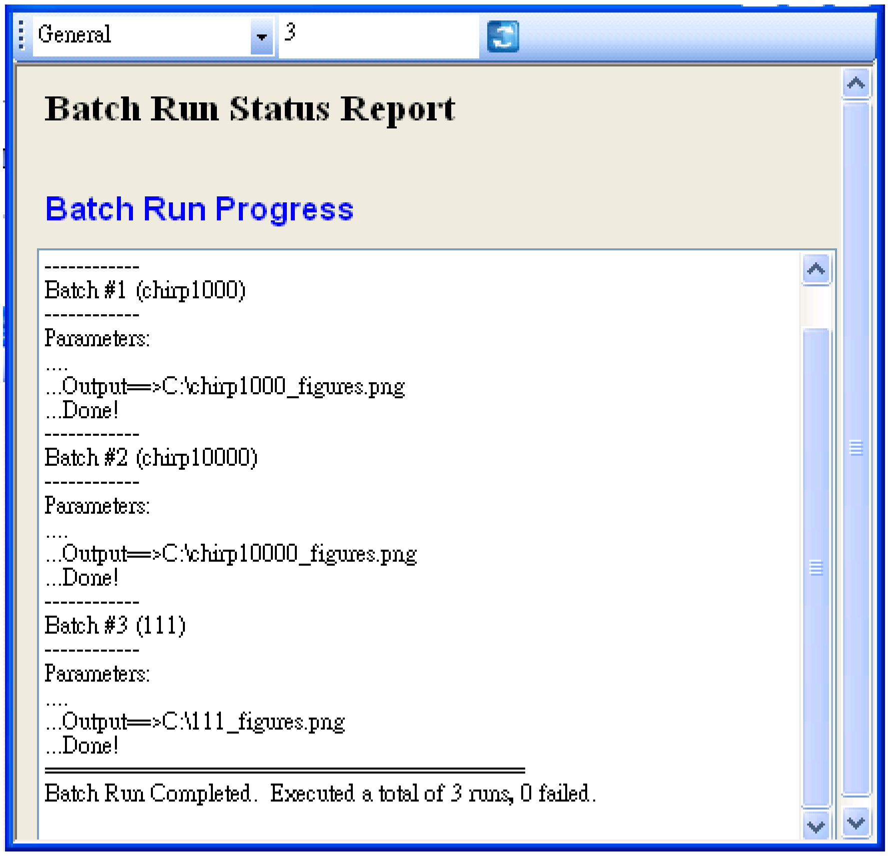


After selecting the DataSource signal files, the Output Directory need to be specified. Click on  to manually locate an Output Directory. In this example, C:\ was chosen.

Click on  to execute the *Batch Run*.



After a *Batch Run*, a Status Report will be generated telling you if the process has been a success or failure.



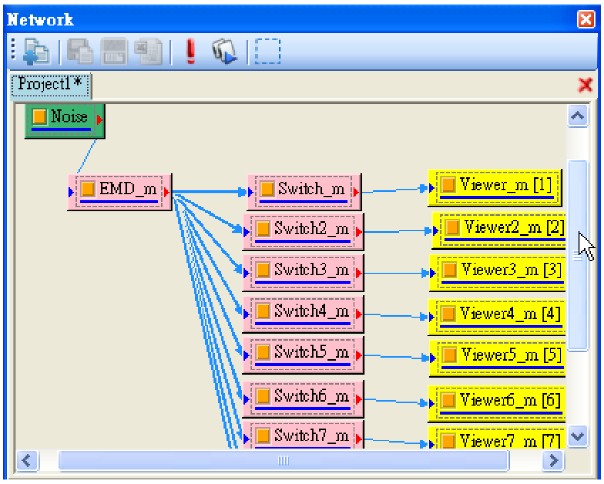
In summary, Example 1 shows how to change the parameters of *Batch Runs* and Example 2 shows how to execute multiple files within the *Batch Run*. You can also run a *Batch Run* which changes the parameters of the variables while executing multiple files.

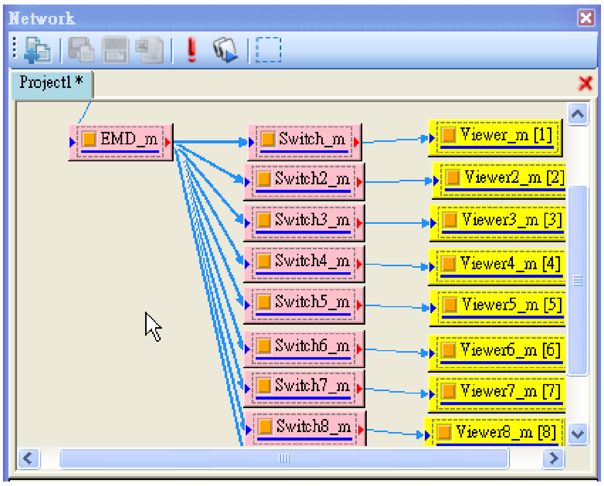
1.3.6 Toggle Selection Mode

Toggle Selection Mode helps user to scroll the view in the Network panel or group SFOs together for copying, cutting, pasting, moving, and deleting.

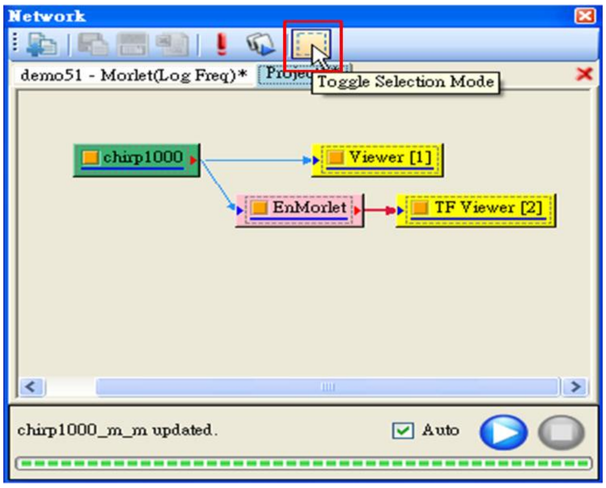
For scrolling display view of the Network panel,  should not be pressed. For example, if there are many SFOs in the Network panel, horizontal and vertical Scroll Bar can be used to change the display.

To move a single SFO, select the SFO and hold the left mouse button in the Network panel, then drag the SFO to the desired location and release the mouse button.

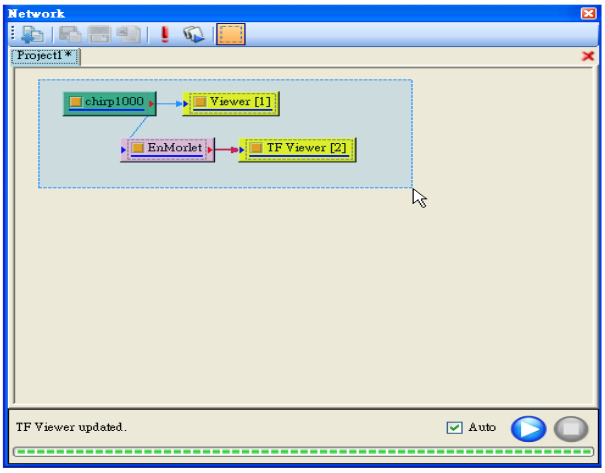




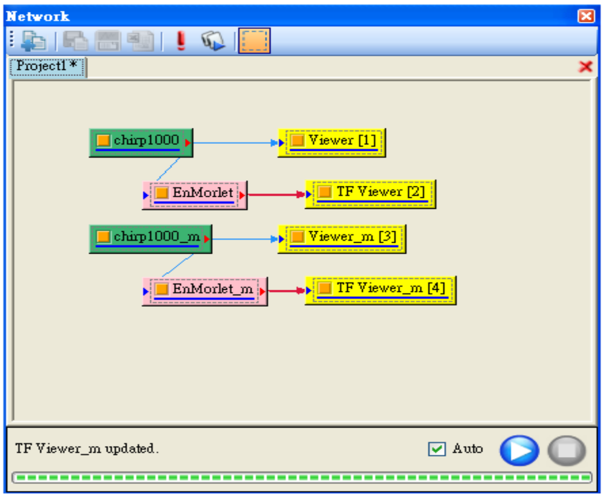
For grouping SFOs,  button must be selected first, then other editing operations can follow.



In the Network panel, press and hold left mouse button, drag the mouse and cover target SFOs, then release the mouse to select them. Press “Ctrl + C” for coping and “Ctrl + X” for cutting, as shown below.

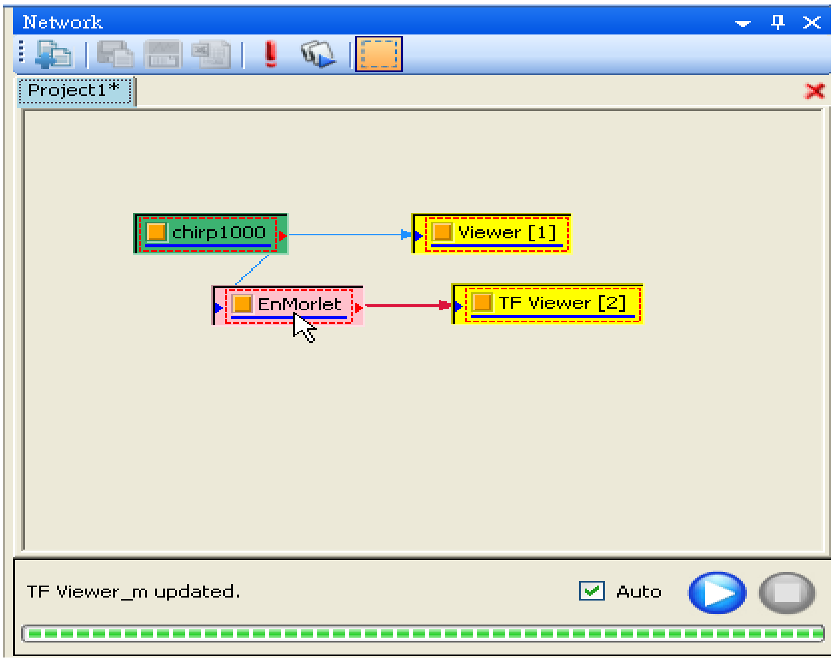


Finally, press “Ctrl + V” for pasting selected SFOs in the Network panel.

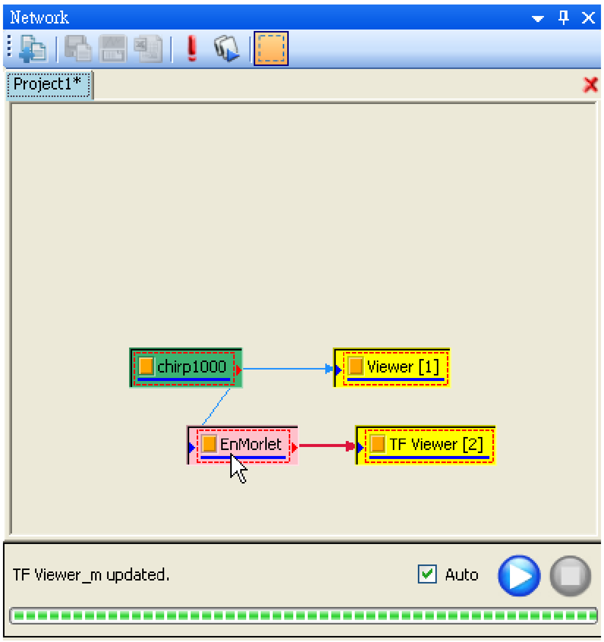


Now let us show how to move SFOs. In the Network panel, press and hold left mouse button, drag the mouse and cover targeted SFOs, then release the mouse to

select them. Hold “Ctrl” key and click/select one SFO via left mouse button, hold the left mouse button and make sure that the group of SFOs is selected.

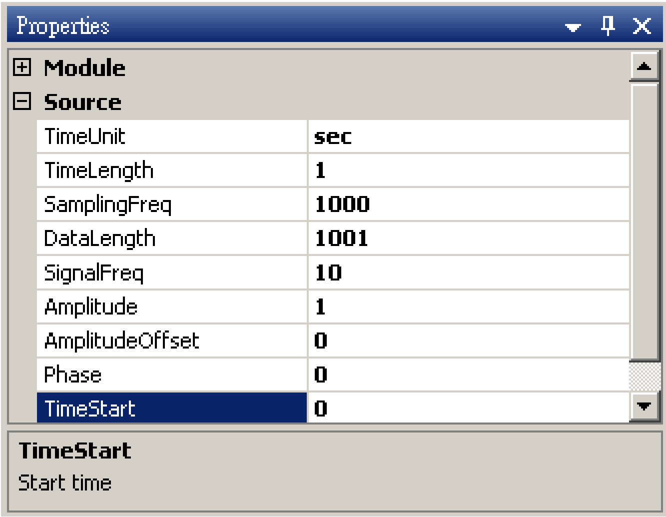


In the Network panel, move the mouse to the target location, then release the left mouse button and “Ctrl” key.

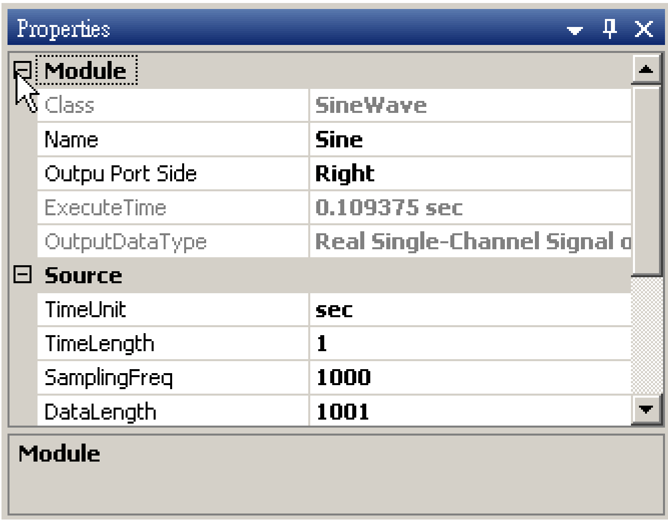


1.4 Properties Window

The values of the SFO in the *Network Workspace* can be edited within the *Properties Window* below the *Network Workspace*. Below is the *Properties Window* of a Sine wave and the options to change the display of the graph can be edited under *Properties/Source*.



In the *Properties Window*, click on [+] to expand the information under Module. *Class* refers to the type of source the SFO belongs to, *Name* shows the name of the SFO and can be renamed, *Output Port Side* allows a change of output lines coming out of the SFO (**TIPS:** Sometimes there is a need to change the sides to better display the entire Signal Flow Diagram) and *Execution Time* shows you the time it took to calculate this information.

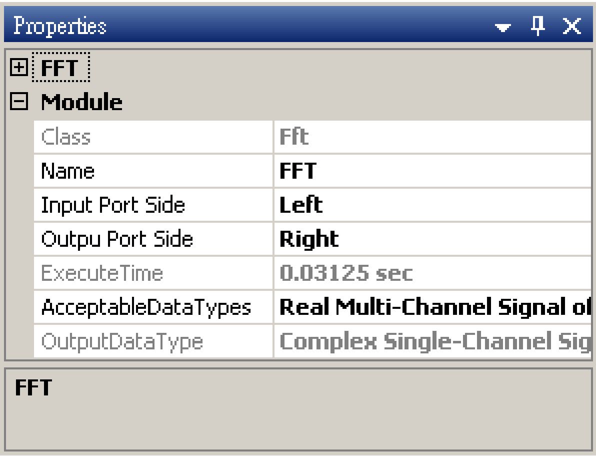


If a SFO contains both input and output function e.g. Compute and Conversion, then there will be a lot more information shown under Module. For example there might be new information such as Input Port Side, Acceptable Data Type and Output Data Type included.

Input Port Side is exactly the same as Output Port Side except Input Port Side determines the position of the lines connects to the SFO.

Acceptable Data Type is a drop-down menu which displays all the acceptable data input that the SFO can be connected to. Selecting an option from the drop-down menu will not perform any action because the drop-down menu is just a display.

Output Data Type displays the output file format.

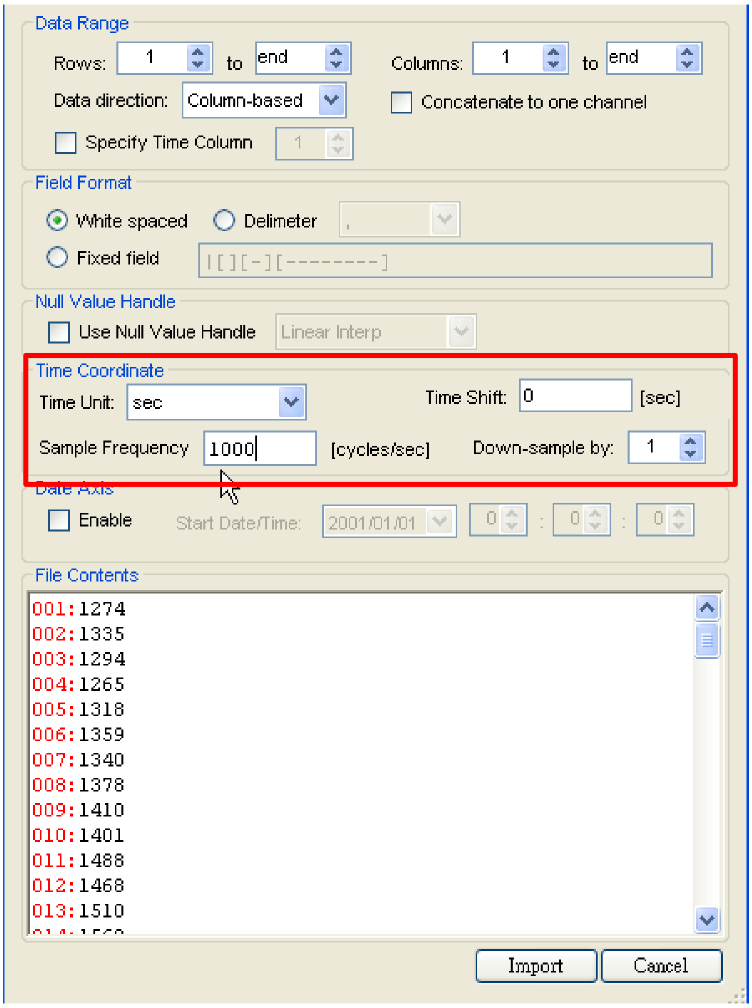


Signal data:

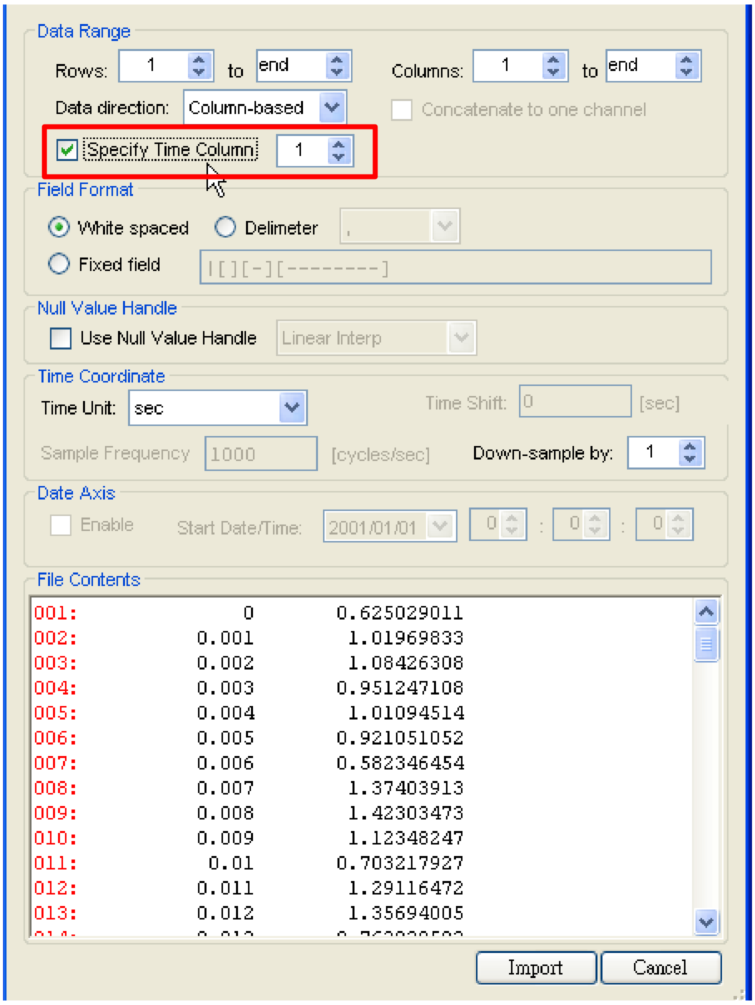
Data	Data Definition
Channel Types	Real: Real Signal Complex: Complex Signal
Channel Number	Single-Channel: Single Channel Multi-Channel: Multiple Channels
Data Types	Signal: Signal Data Audio: Audio Data Numeric: Numeric Data

	Spectra: Spectra Data
Information Types	Regular: Equal distance between points. Indexed: Irregular distance between points.

Regular Data and Indexed Data often cause the most problems for users so further explanation on these two types of data are required. Regular Data means that the points on the x-axis have the same distance between each other, so the increment from one point to next point is always the same.

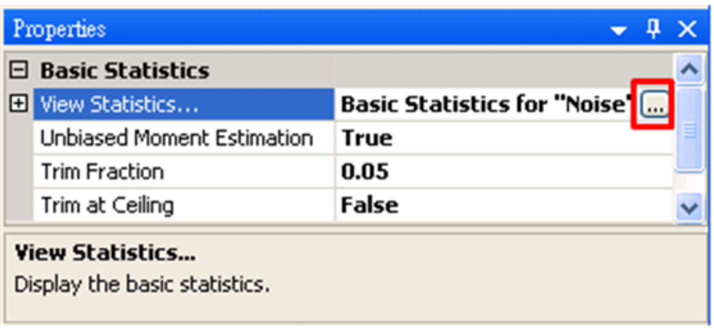


In the Indexed Data type the distance between each point along the x-axis is irregular. If the imported file contains time information (does not matter whether it is regular data type or indexed data type), then check the *Specify Time Column* in the Text Importer and the Text Importer will assign the first column to be the time column for the data.



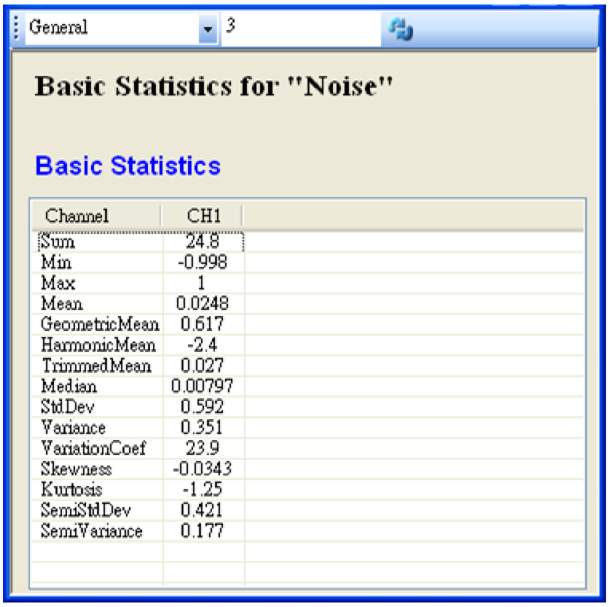
Reporter Window:

Reporter Window is only available when the output data type is numeric. Signal Flow Objects, such as Basic Statistics, Correlation Matrix, Covariance Matrix, Orthogonal Matrix, Qurtiles, and Quantiles, can open the Reporter Window to displays statistical information based on the data calculation. The Reporter Window can be accessed through clicking on the button on the *Properties/View Statistics*.



There are two options and a refresh button at the top of the Reporter Window.

The first option is the way the decimal numbers are displayed. The default is *General* decimal display and in the drop-down menu there are also *Fixed* decimal display and the *Scientific* decimal display. The second option is to configure how many decimal places that will be displayed and the default is 3 decimal places. Click on the *Refresh* button when either of the options has been changed and the calculations will be updated with the new display settings.

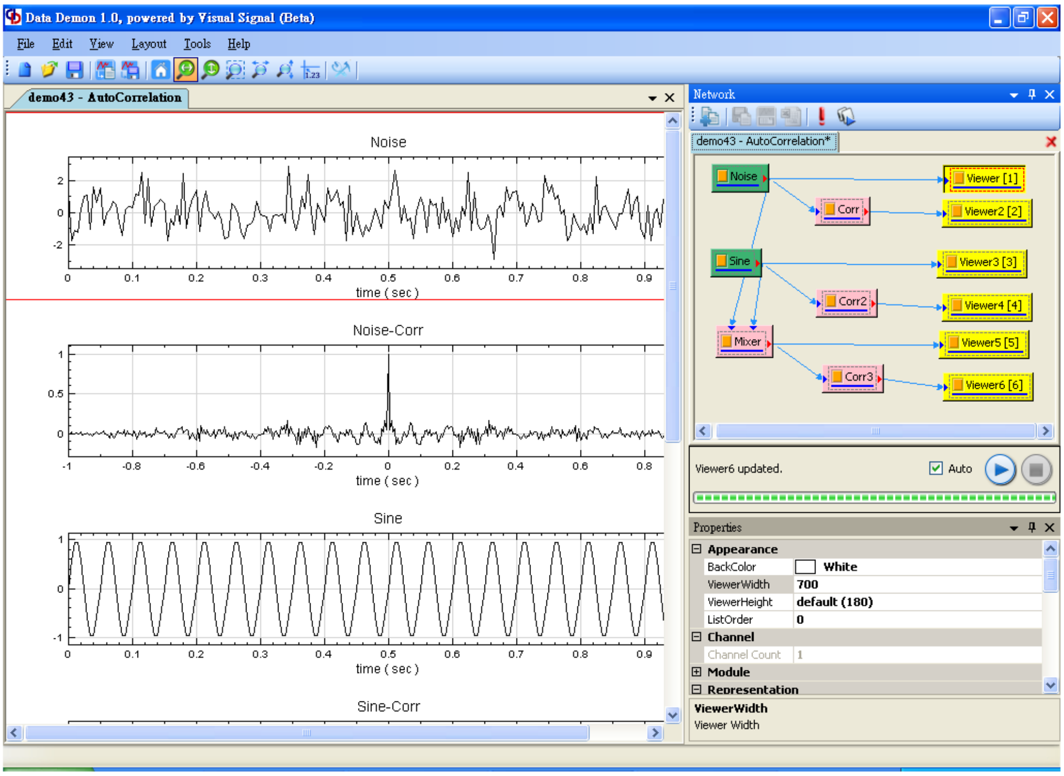


Property Name	Property Definition
General	Statistics generated by the program.
Scientific	Scientific notation, e.g. 1.234E-001
Fixed	Fixed point-notation. Displaying numbers behind the decimal point.

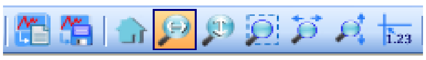
Below are the Signal Flow Objects which contain Report Window.

Reporter Component Option
IMF Property
DoMatlab
Correlation Matrix, Covariance Matrix
Orthogonality Matrix
Quartiles and Quantiles
Batch Run

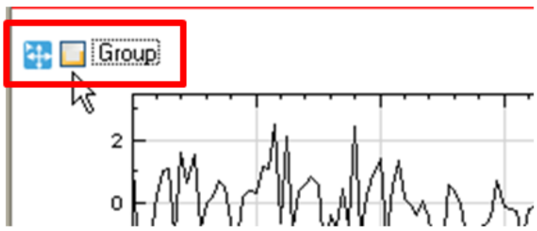
1.5 Visualization Window



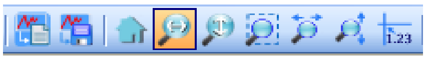
When Viewer SFO is created, a graph corresponding to the SFO source will be shown on the *Visualization Window*. The graph of the firstly created Viewer SFO will be shown as the first graph, and the graph of the secondly created Viewer SFO will be shown as the second graph and so on. DataDemon provides many useful functions to control, display, and export the graphs shown in the *Visualization Window*.



The *Visualization Window Toolbar* is located under the menu toolbar at the top left corner of DataDemon.



1. Visualization Window Toolbar



Copy to Clipboard allows you to copy the graph into the clipboard to be pasted into another application (**Note:** Copy to Clipboard provides two types of file format, Bitmap and Meta file). *Export File* allows you to save the graph into variety of file formats such as PNG, BMP, JPEG, TIFF, and WMF etc.

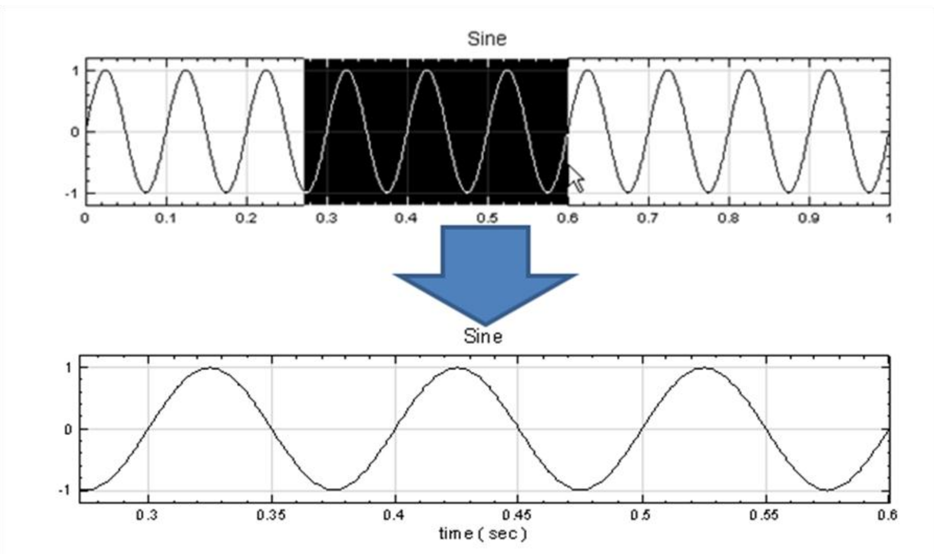
Besides the first two buttons and the last button, the rest of the buttons on the tool bar will directly affect the graph displayed in the *Visualization Window*.

(1). Home

Clicking on the *Home* button will reset the graph to its default position and size. This button is useful when you get lost with zooming and moving around the graph and want to set the graph to its default view.

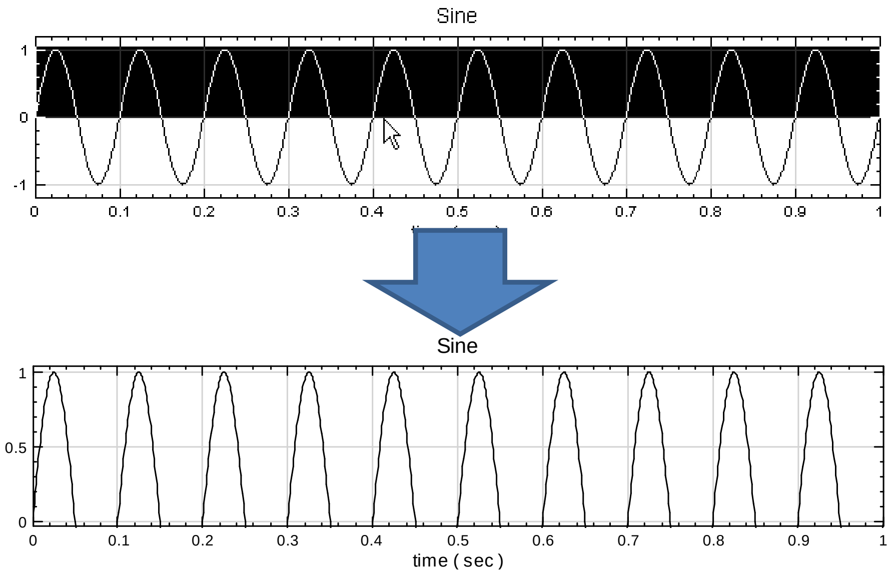
(2). Zoom X

Firstly, click on the *Zoom X* button and then click on any part of the graph and drag it along the x-axis. Part of the graph is highlighted. As you release the mouse button after dragging, the highlighted area will be zoomed in and displayed.



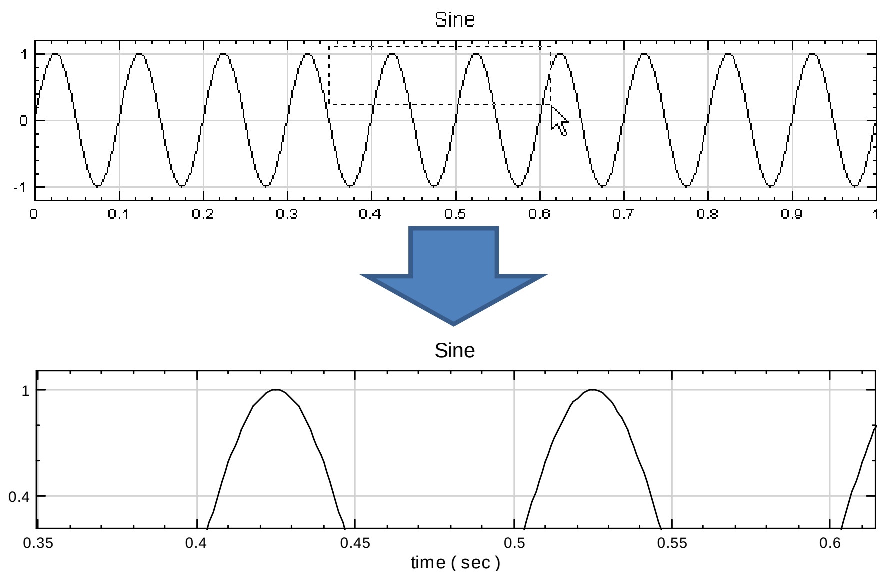
(3). Zoom Y

Firstly, click on the *Zoom Y* button and then click on any part of the graph and drag it along the y-axis. As you release the mouse button after dragging, the highlighted area will be zoomed in and displayed.



(4). Zoom Rect

Firstly, click on the *Zoom Rect* button and then click on any part of the graph and drag a box. As you release the mouse button after dragging, the selected box area will be zoomed in and displayed. This function combines the affect of *Zoom X* and *Zoom Y*.

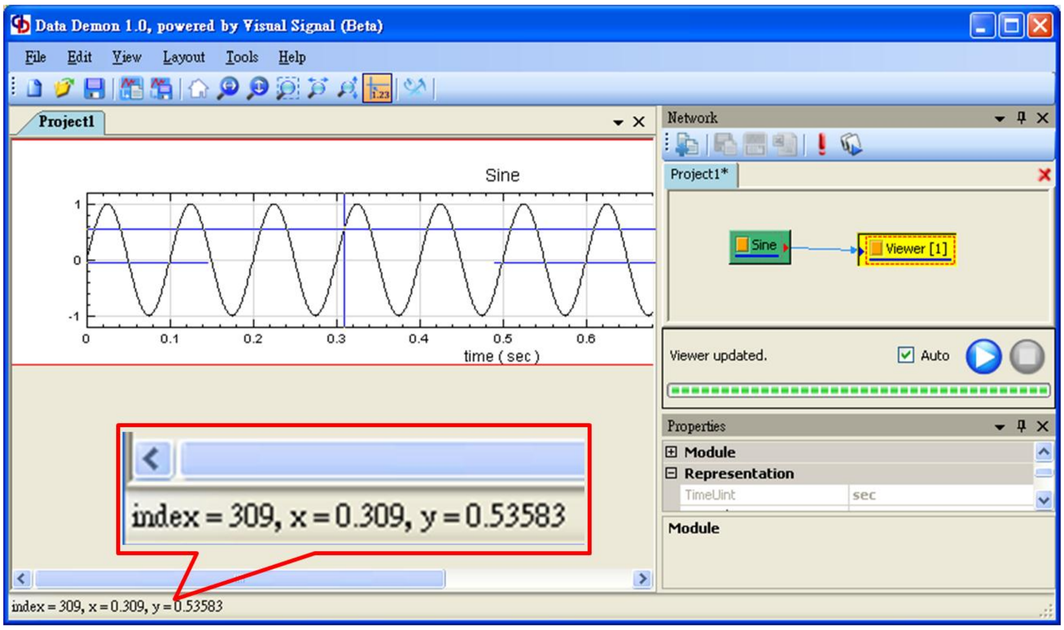


(5). Pan X, Pan Y

Pan X and *Pan Y* buttons allows you to move along the x-axis and the y-axis of the graph respectively.

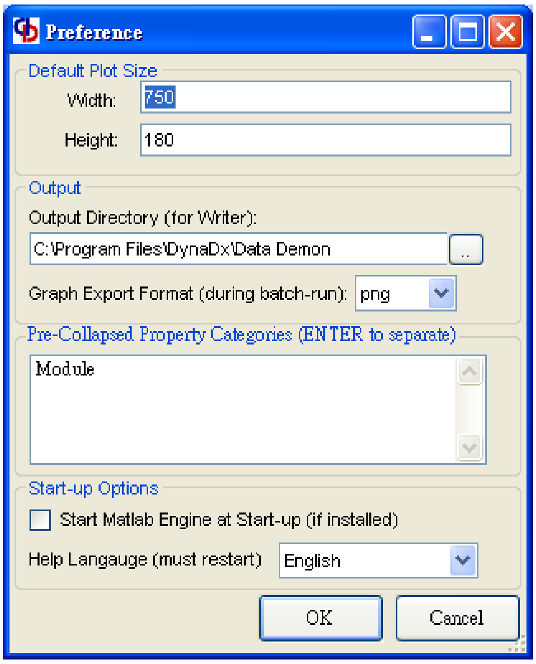
(6). Show Value

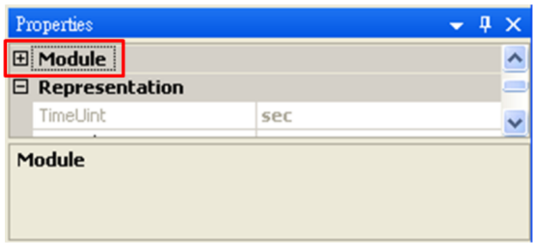
Firstly, click on the *Show Value* button and then move the mouse onto any part of the graph. There will be a horizontal and a vertical blue line which will intercept a data point on the graph and the value of that point will be displayed at the bottom left corner of DataDemon program.



(7). Preference

Preference allows you to edit some configuration regarding DataDemon. *Default Plot Size* allows you to change the graph dimension in the *Visualization Window*. *Output Directory* sets the Writer SFO's default output location when saving a file. *Graphic Export Format* allows you to set the types of image file format that will be saved during a *Batch Run*. *Pre-Collapsed Property* allows the category name entered to be compacted showing a [+] next to the category name in the *Properties Window* (as shown in the image below).

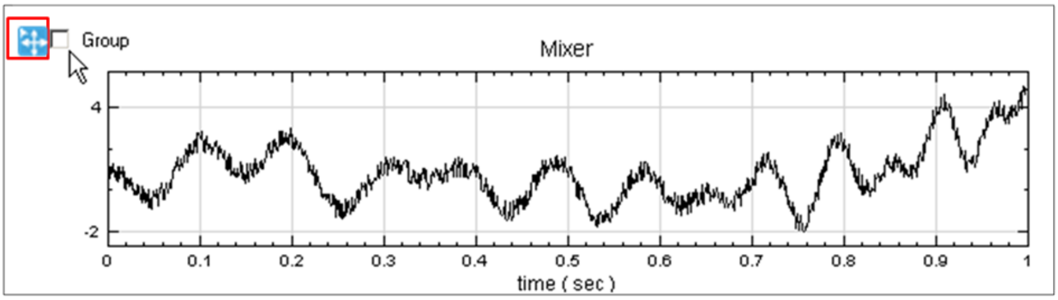




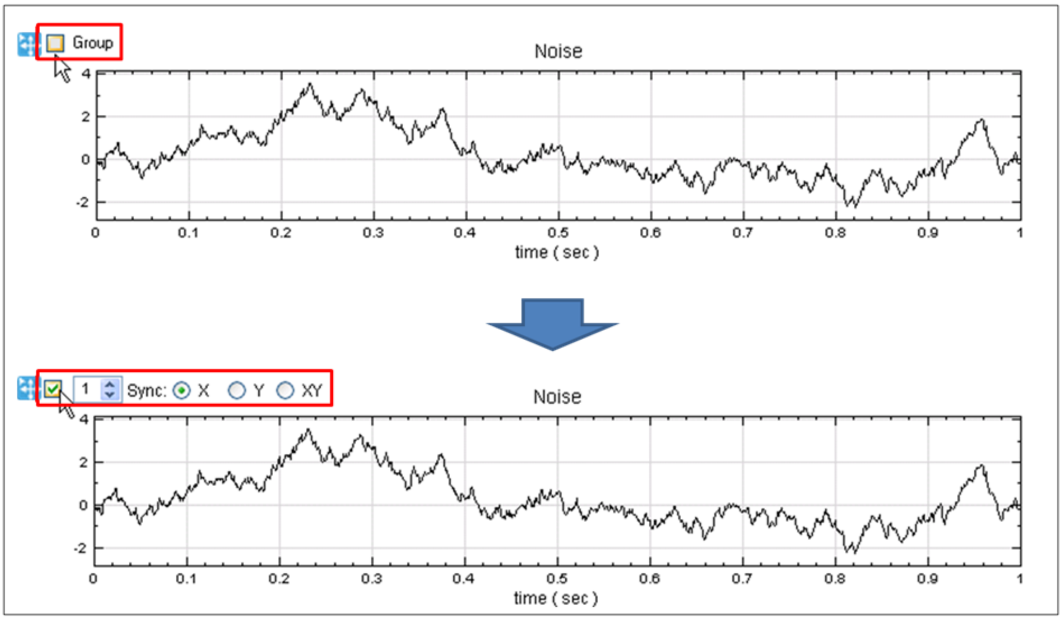
There are two configuration listed under *Start-Up Options*. The first option is the *Start Matlab Engine at Start-Up* and the second option is the *Help Language* which will support a range of language (**NOTE:** Currently only English is supported).

2. Image Toolbar

There is a hidden tool bar on the top left corner of every graph. Move your mouse cursor to the top left corner of a graph and the hidden image toolbar will be displayed. The image toolbar is used to synchronize different graphics together so that they can be zoomed or moved with the same increment. The other feature is to move the graph up or down (providing there is more than one graph). Just click on  and then drag it to the top graph or the bottom graph (**NOTE:** You can only move one position at the time).

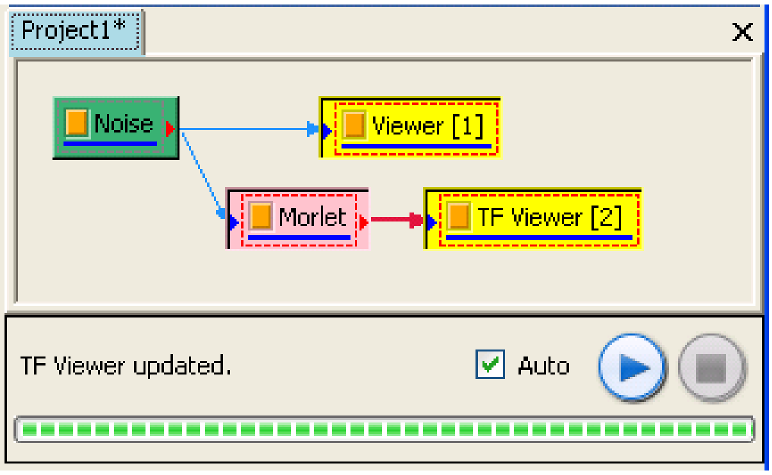


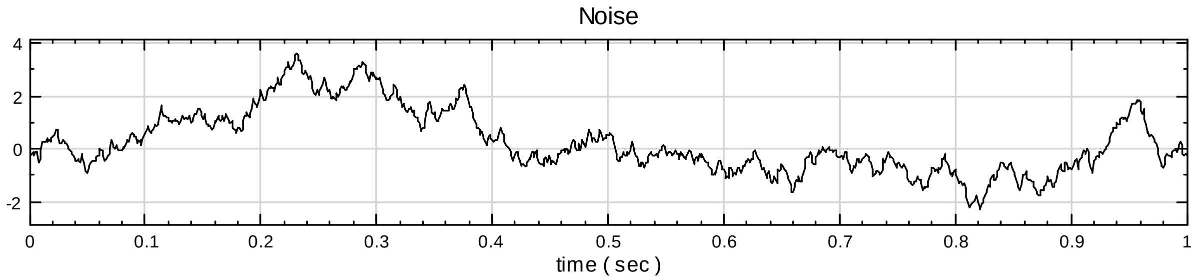
Next to the is the *Group* tick box, it is used to synchronize many images so that they can be zoomed and moved together. Check the *Group* tick box to start synchronizing some images. A small scrolling bar allows you to enter a number between 1 to 5 and this number indicates the group that the image will be associated with. The images belong in the same group will be zoomed and moved with each other. Sync option X, Y and XY indicates which x-axis or y-axis or both directions will be synchronized and moved.

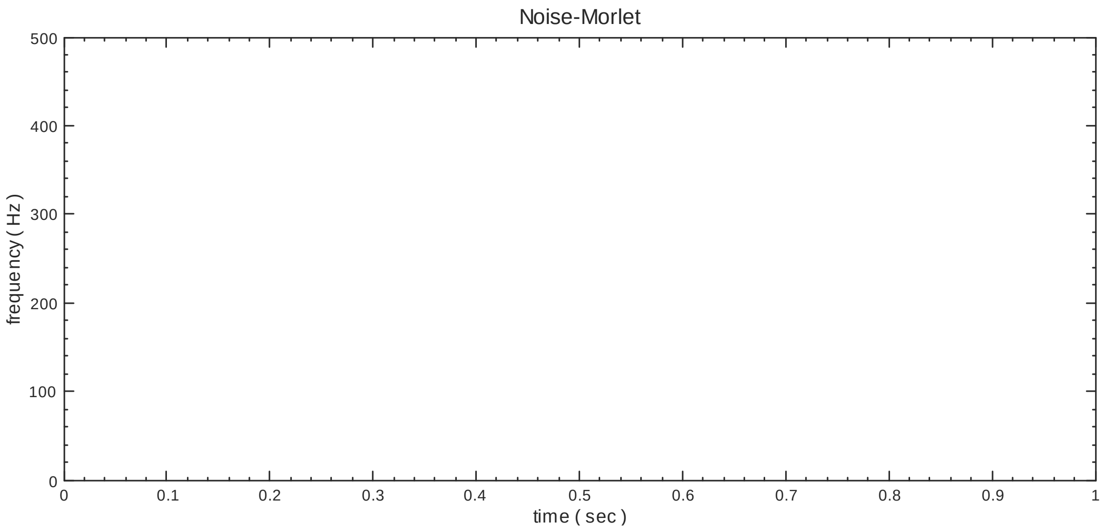


After configuring the Group option, the configured setting will remain visible on the top left corner of each image.

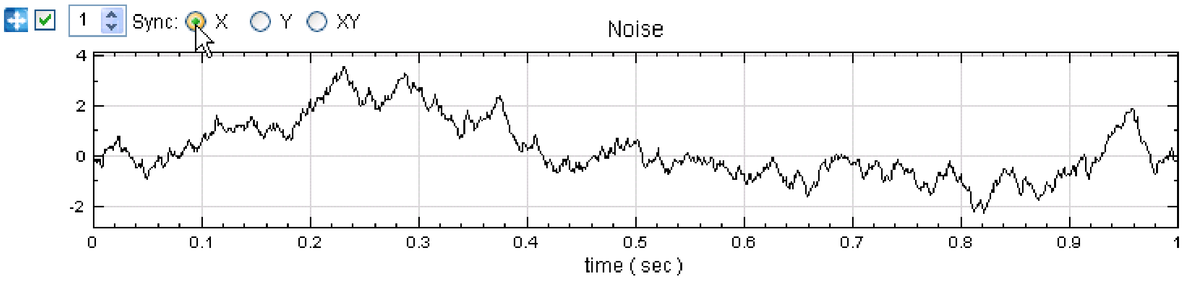
Below is an example of how to use *Group* to synchronize two images so that they can be zoomed and viewed together. The Signal Flow Diagram is made up of *Source→Noise* and viewed with *Viewer→Channel Viewer* and the Noise SFO is also connected to *Compute→TFA→Morlet Transform* and viewed with *Viewer→Time-Frequency Viewer* (as shown in the image below).


Project1*
Noise
Viewer [1]
Morlet
TF Viewer [2]
TF Viewer updated.
Auto

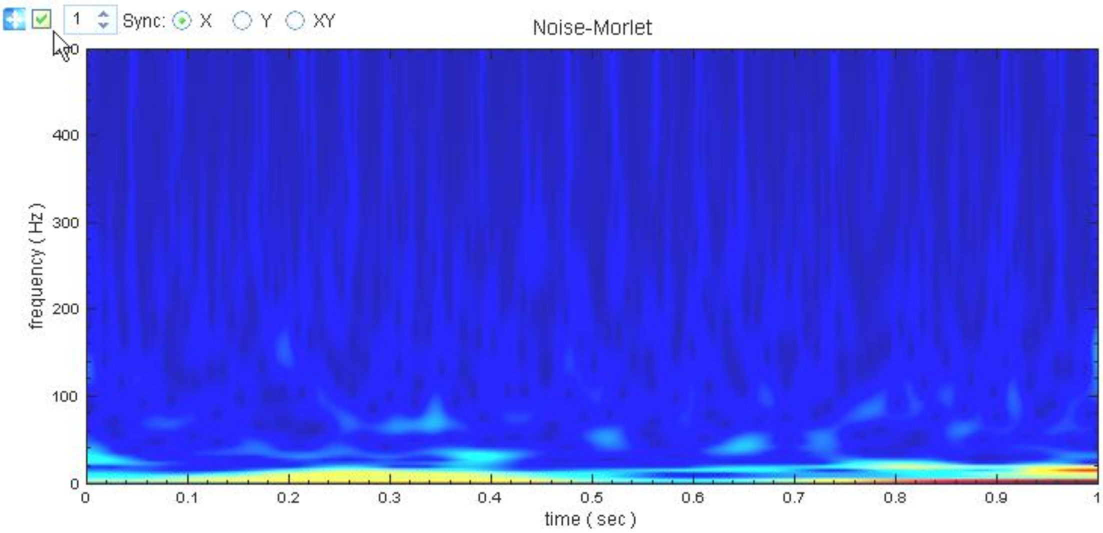

Noise
time (sec)


Noise-Morlet
frequency (Hz)
time (sec)

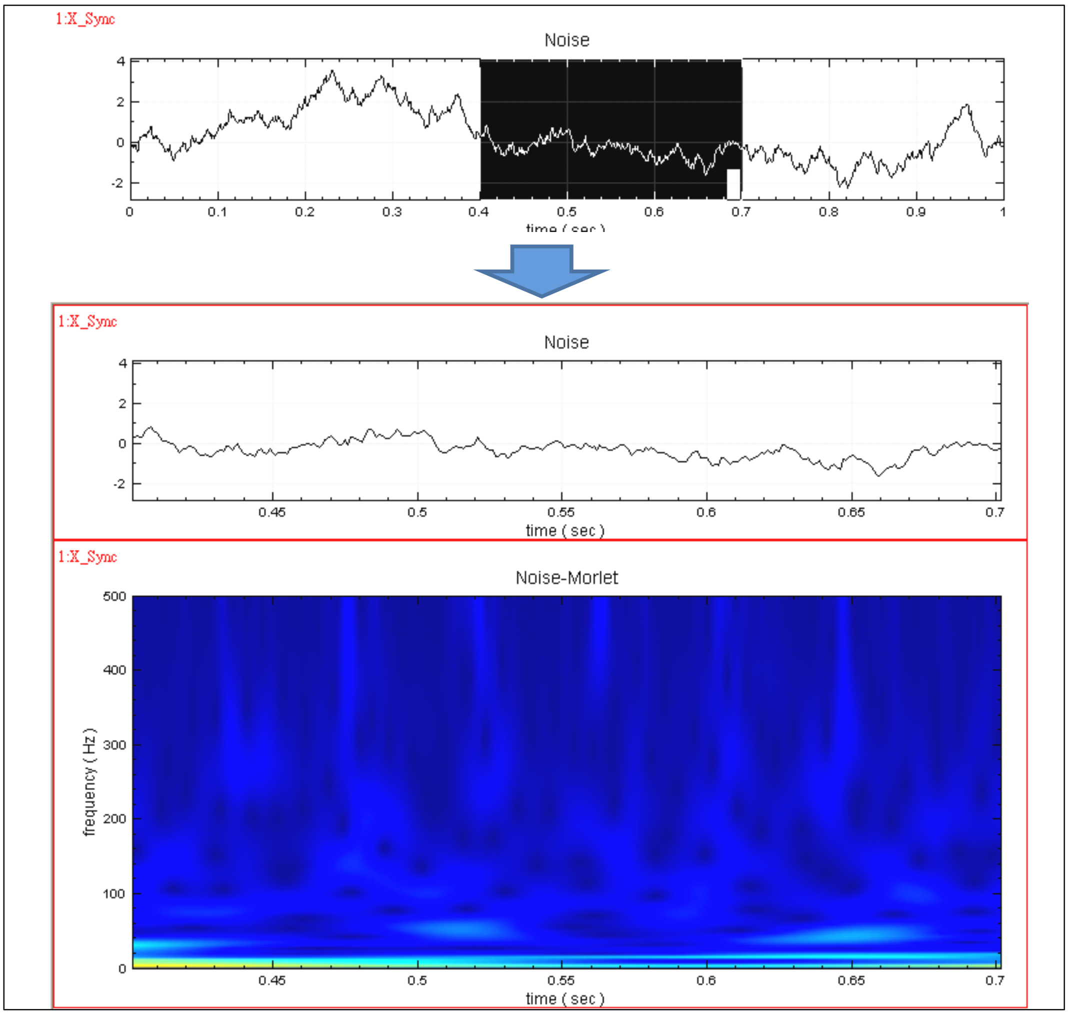
Move the mouse cursor to the top left corner of the top graph (first graph) and the *Group* tick box will appear. Check the *Group* tick box, leave the number as 1 and then select *Sync X*.



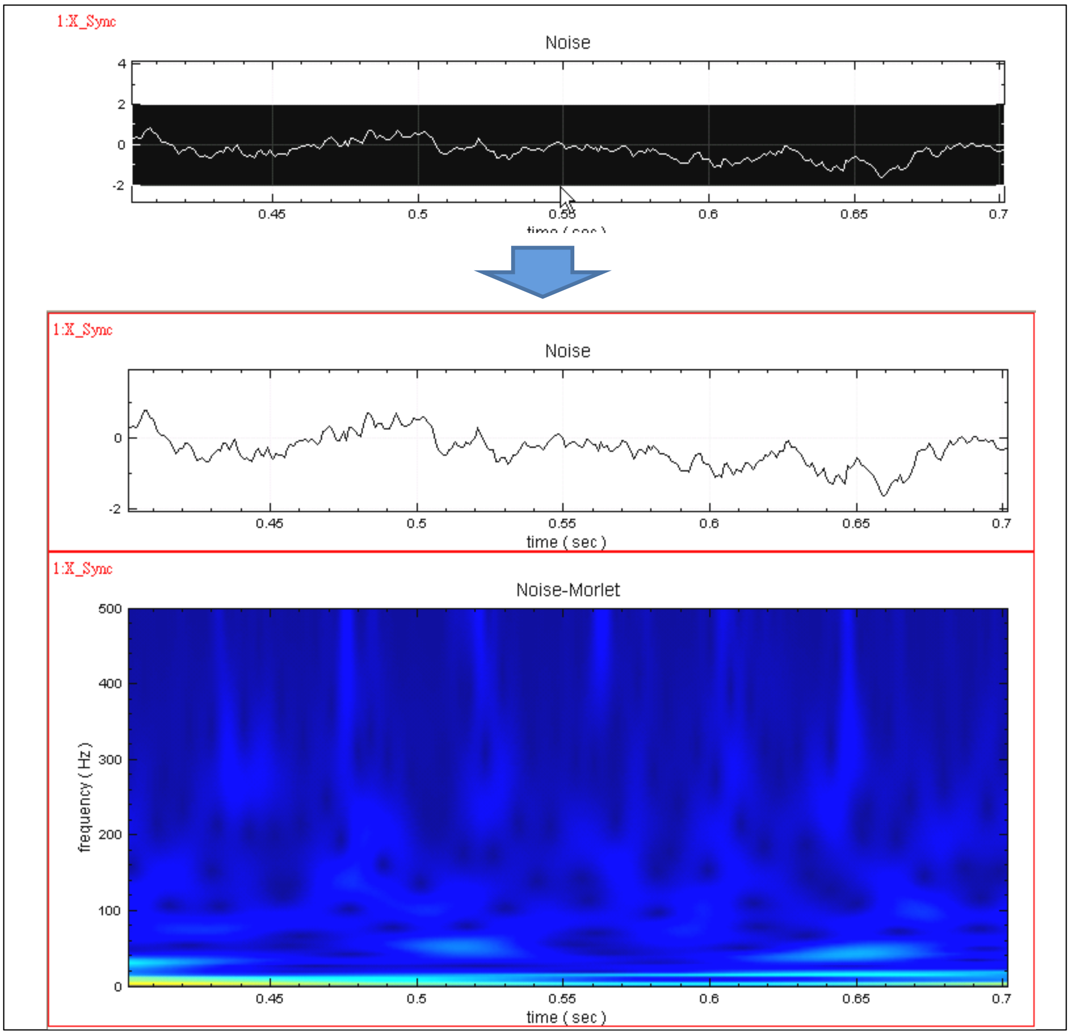
Now repeat the same steps done to the top graph and apply it to the bottom graph (second graph). Move the mouse cursor to the top left corner of the bottom graph and the *Group* tick box will appear. Check the *Group* tick box, leave the number as 1 and then select *Sync X*.



To demonstrate the usefulness of synchronization, select *Zoom X* from the *Visualization Window Toolbar* and drag the mouse from 0.4 sec to 0.7 sec on the x-axis. You will see that both graphs will be zoomed into the 0.4 sec to 0.7 sec area.

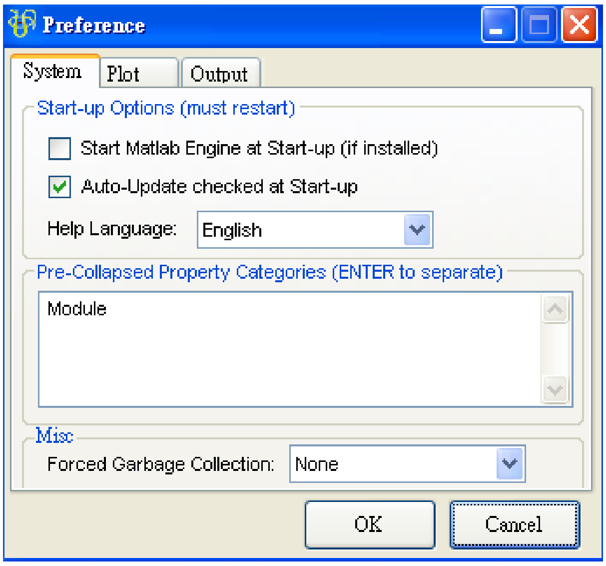


Using *Pan X* on one of the graph will move both graphs along the x-axis. The above example uses *Sync X*, so toolbar features involving x-axis can alter the graphs. But if you wish to move in the y-axis direction or move in the x-axis and y-axis direction then you can select *Sync Y* or *Sync XY* instead. (Shown in the image below is an example of *Zoom Y*).



1.6 Preference

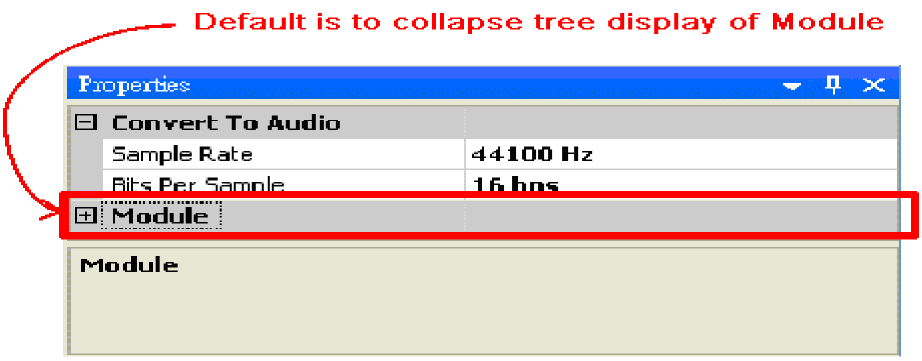
Preference sets the default parameters for the program. It contains three sections: System, Plot, and Output.



System Tab

In the System / Start-up Options section, we can a) select to start Matlab Engine (if Matlab is installed) when DataDemon starts; b) select to check upgrades automatically; c) select language interface for DataDemon, currectly only English and Traditional Chinese are supported. The program must restart after the changes of these options.

In the System / Pre-Collapsed Property section, we can set the Properties content format of each SFO, the default value is “Module”. It means that the Module property in Properties of each SFO is in collapsed status when initial displayed as shown below.

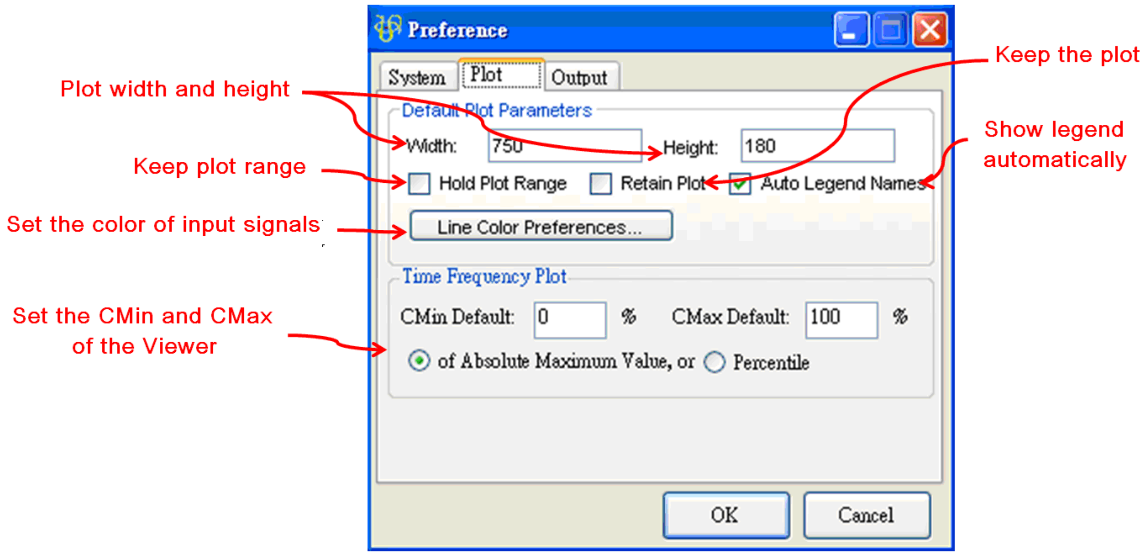


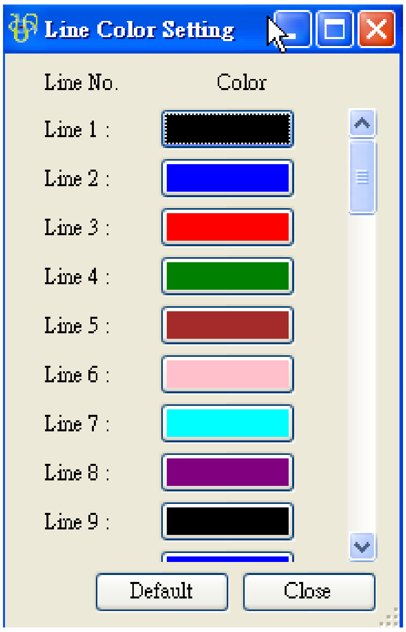
Forced Garbage Collection sets when to do the gargabe collection: None, Per Project, and Per Module. It helps to release memory space.

Plot Tab

In the Plot / Default Plot Size section, we can a) set the default width and height of the plot in the Visualization panel; b) Hold Plot Range determine whether to keep XY range of the plot or not when setting Viewer SFO parameters; c) Retain Plot keeps the original plot when Viewer SFO is not connected; d) Auto Legend Name sets the legend automatically; e) Line Color Preferences sets the color of each channel in the Viewer SFO as shown the image below.

In the Plot / Time Frequency Plot section, we can set default values of CMin and CMax for the plot, and they can be based on the percentage of the Absolute Maximum Value or Pencentile of the time-frequency data.



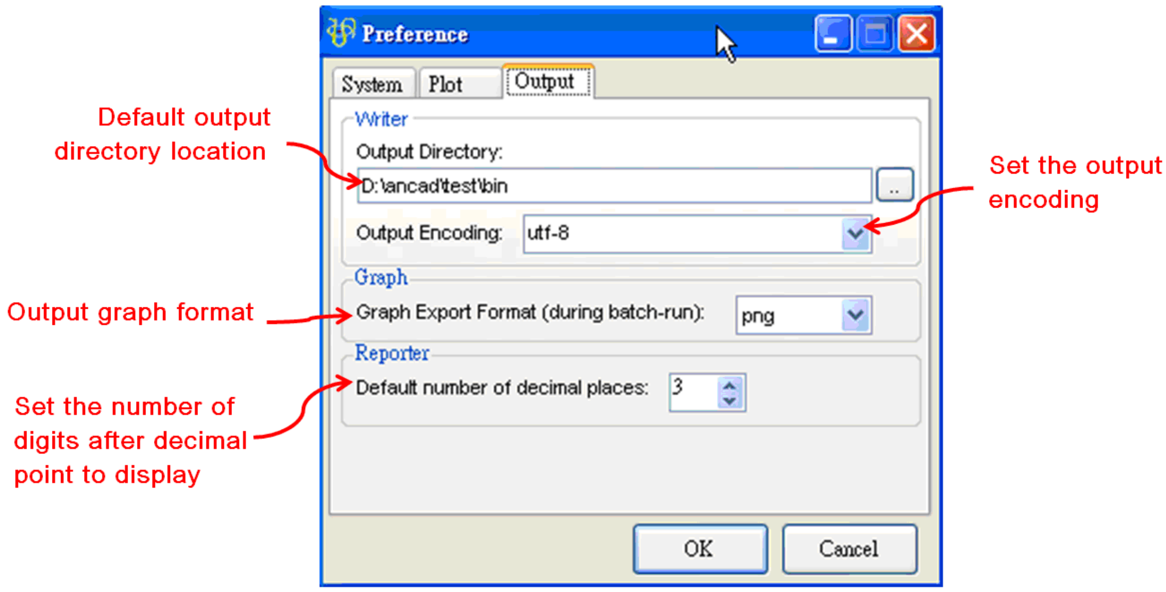


Output Tab

In the Output / Writer section, we can a) set the default outout directory; b) the encoding format of the output file.

In the Output / Graph section, we can set the graph export format during the batch run.

In the Output / Reporter section, we can set the resulotion of the output. The default number of decimal places shows the digits after the decimal point.

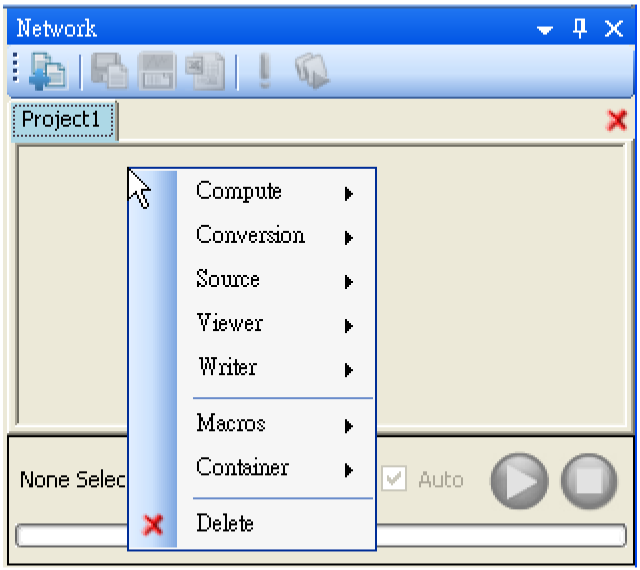


Chapter 2: Network Window

This chapter explains different types of Signal Flow Objects and how to use them.

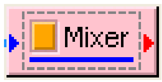
2.1 Signal Flow Object

Right mouse click on the *Network Workspace* of the Network Window will open up the menu (*Network Workspace Menu*) for the Signal Flow Objects. Signal Flow Objects are categoried into five types: *Compute*, *Conversion*, *Source*, *Viewer* and *Writer*. Within Compute there are more sub-categories such as *Channel*, *Filter*, *HHT*, *Mathematics*, *TFA* and *Transform*. All the SFO types will be explained in this chapter.



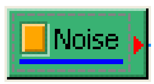
2.1.1 Signal Flow Object Types

1. **Compute:** Compute is represented as a pink colored SFO. It provides different type of signal calculation to the source data/SFO. With Compute type SFOs, the original Source type SFO does not need to be edited and lots of manipulation can be done directly to a single Source SFO. Compute type SFOs can be created to apply special calculations to the data of the Source SFOs without altering the original source data.



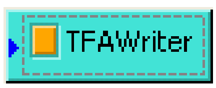
2. **Conversion:** Conversion is represented as a light brown colored SFO. Conversion provides the Source SFO a variety of options to change or convert, such as change its x-axis unit, convert signal data format to playable audio data format, and convert index format data to regular format data etc. Conversion type SFOs can be created to manipulate the data of the Source SFOs without editing the original data.

3. **Source:** Source is represented as a green colored SFO. You can load an external data file or generate a customized wave, noise wave, sine wave, triangle wave and square wave from this type.



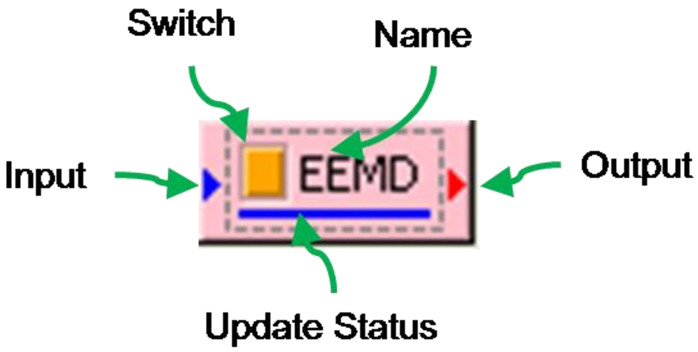
4. **Viewer:** Viewer is represented as a yellow colored SFO. It is used to display graphs and images from source signals or manipulated signals (such as the ones that have gone through changes from Compute and Conversion types).

5. **Writer:** Writer is represented as a light blue colored SFO. It is used to output the calculation and data from a SFO to a file with specified text or sound formats.

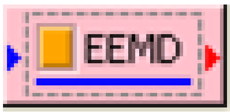


2.2 Components of a Signal Flow Object

Image below is a Signal Flow Object and the explanations of each component.



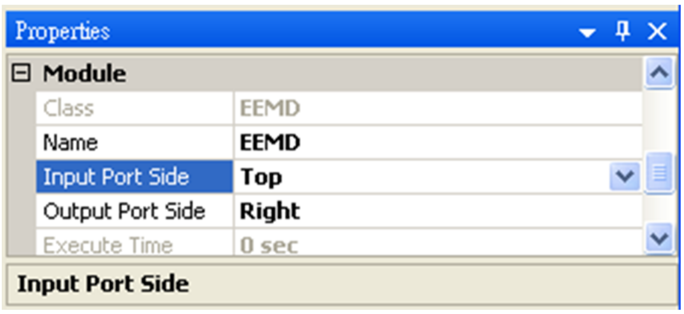
2.2.1 Input, Output and Name

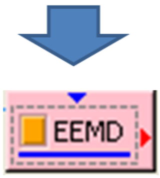


1. **Input:** Shown in the image above is a SFO. The blue triangle to the left of the SFO is the input port. The EEMD SFO will receive data from other SFOs. The number of input ports varies depending on the SFO, e.g. Merge To Complex SFO has two input ports.

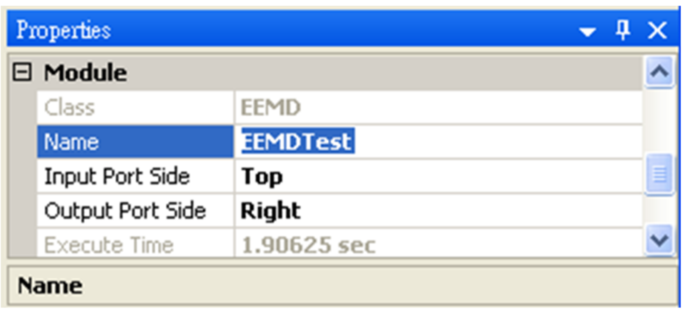
2. **Output:** The red triangle to the right of the SFO is the output port. In this example it will transfer the data generated by the EEMD out to its output port. All SFO's output port can be connected to one or more SFOs.

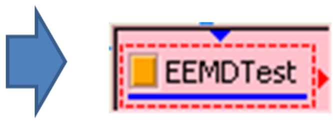
   The position of both Input port and Output port can be changed within the *Properties/Module* of a SFO. In the example below, *InputPortSide* (Input) is changed to *Top* and now the blue triangle is moved to the top.





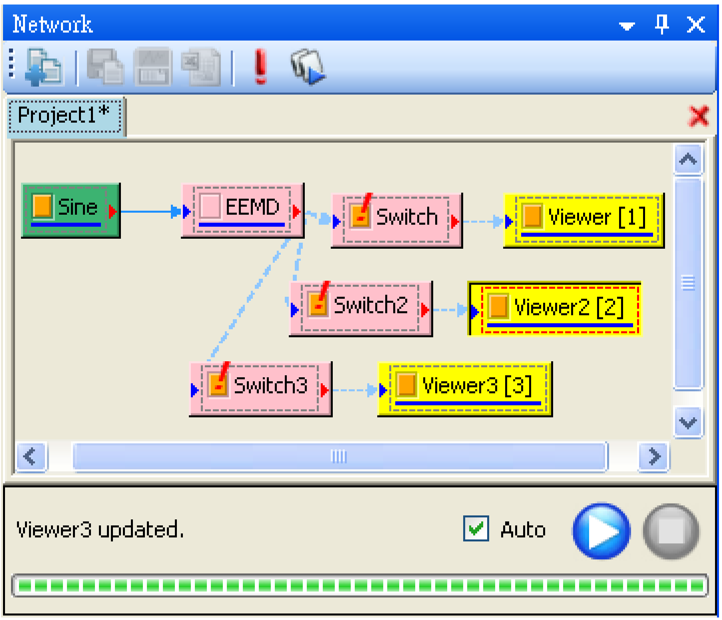
3. **Name:** The name EEMD is created by default when you select EEMD from the *Network Workspace Menu*. Edit the *Properties/Name* to change the *Name* field.



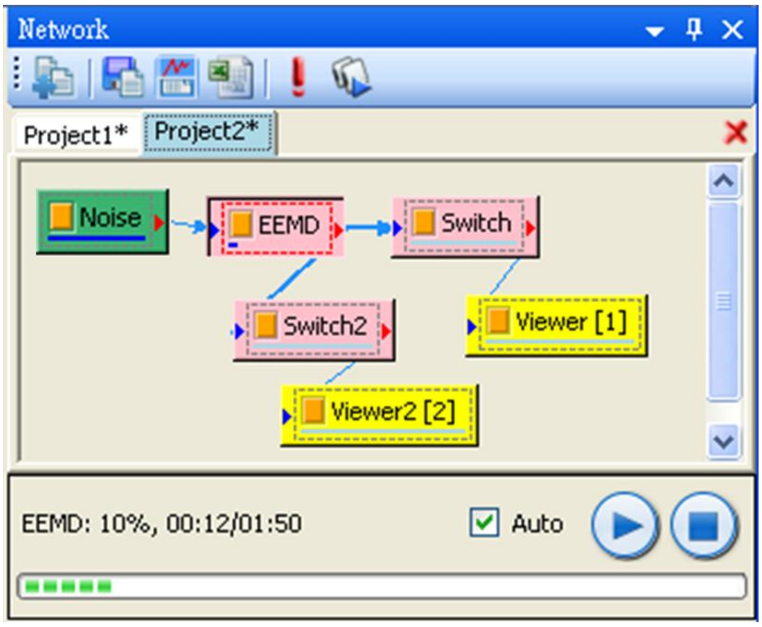


2.2.2 Status Update and Control

There is a small square box located in front of the name of the Signal Flow Object. This box indicates if the SFO is currently active or not. If the SFO is active, it should be displaying an orange colored box. Active means that the SFO is currently calculated and it is connected to the network. When the orange box is clicked, it becomes a transparent box indicating that the SFO is currently inactive. Inactive means that the SFO is not calculated and it is disconnected from the network. An inactive SFO will not output data into any connecting SFOs.



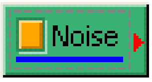
The Update Status has three status and they are visible by looking below the name of the SFO. Empty status means that the SFO is newly created and haven't been updated yet. There is no line shown below the name of the SFO in an empty status. Updated status means that the SFO have been updated and the status is displayed by a dark blue line below the name of the SFO. Outdated status means that the SFO have previously been updated but now it is outdated and is in need of an update. The outdated status is displayed by a light blue line below the name of the SFO.


Network
Project1*
Project2*
Noise
EEMD
Switch
Switch2
Viewer [1]
Viewer2 [2]
EEMD: 10%, 00:12/01:50
Auto

2.2.3 Input and Output SFO Types

There are three types of input and output SFO. They are the input only SFOs, output only SFOs and the SFOs which contains both input and output.

1. **Output only SFO:** This type of SFO consists of output port only. SFOs such as Source and *Viewer→Annotation* are all output only SFOs.



2. **Input only SFO:** This type of SFO consists of input ports only. SFOs such as Viewer and Writer are all input only SFOs.

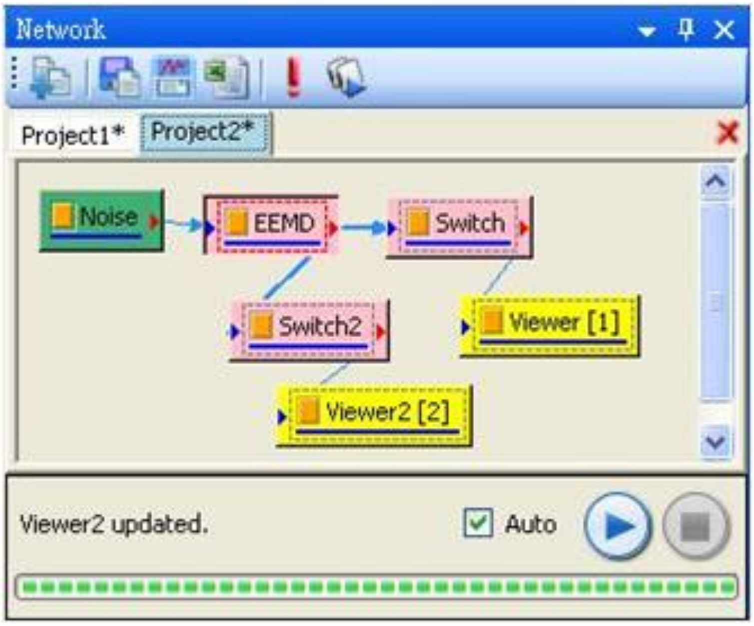
3. **Input and Output SFO:** This type of SFO consists of both input and output ports. It is able to accept data through its input port and process it and send it out through its output port. SFOs such as Conversion have both input and output ports.

2.3 The Usage of Signal Flow Objects

Controlling Signal Flow Objects, how to select, configure and connect Signal Flow Objects in the *Network Window*.

2.3.1 Signal Flow Object Status

Click on a Signal Flow Object to select it. When a SFO is selected, the SFO box in the Network Window will look like the box is pressed in. The *Properties Window* will display the information of the current selected SFO. You can hold down *Control* on the keyboard and select multiple SFOs to move them around the *Network Workspace*.



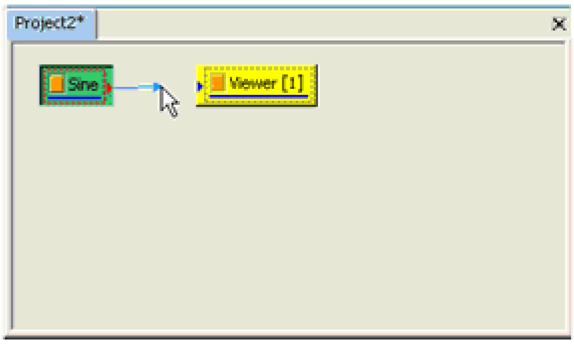
2.3.2 Connecting Signal Flow Objects

With Signal Flow Diagram, DataDemon simplifies the tedious process in analyzing signal data. All you have to do is to create simple Signal Flow Diagram and connect Signal Flow Objects together. The the signal then is analyzed and the result is caculated.

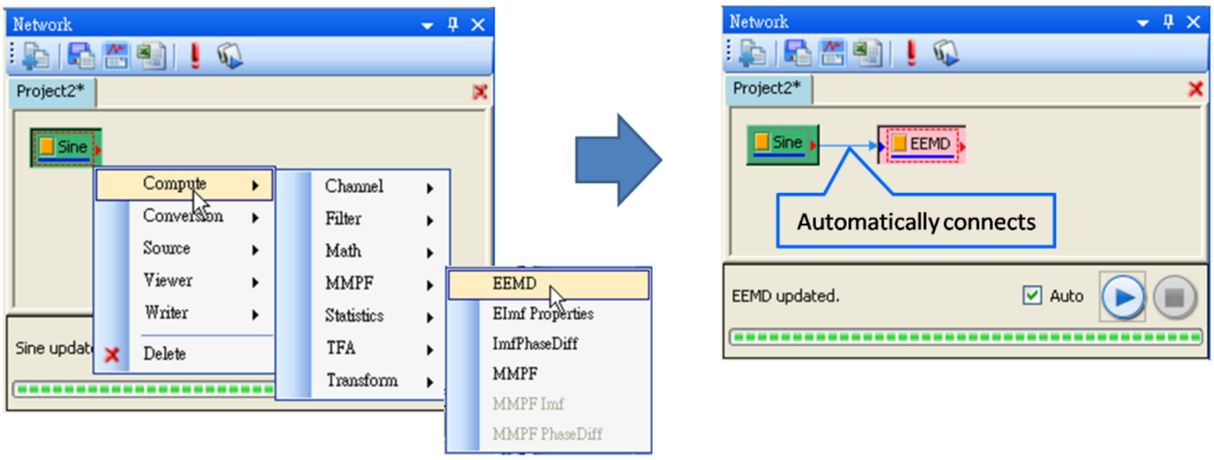
1. To connect two or more Signal Flow Objects.

There are two methods to connect SFOs together. The first method is to create two SFOs and click on the output of one SFO and drag it to the input of the other SFO. The second method is to create a SFO and then right mouse click on it and select a second SFO from the *Network Workspace Menu*. The SFOs will then be automatically connected together.

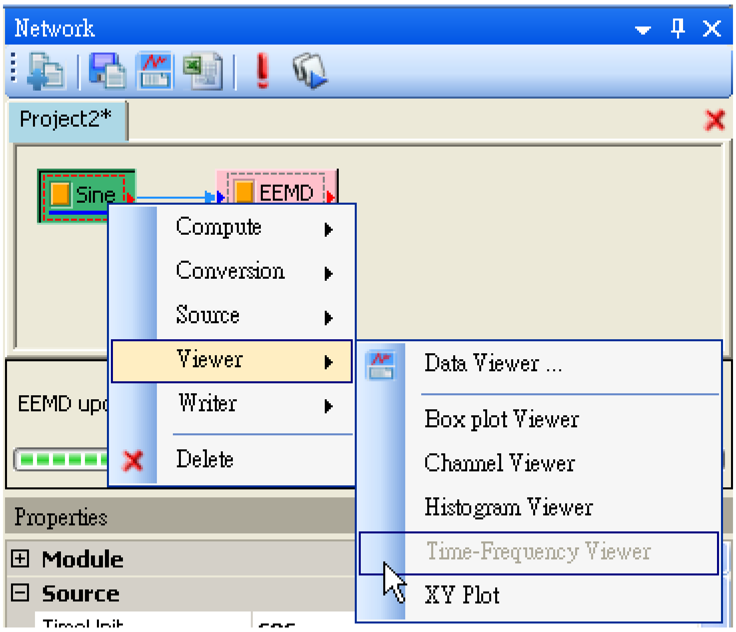
- Mouse click drag the output port of the Sine SFO to the input port of the Viewer SFO



- Select the Sine SFO and right mouse click to select EEMD SFO from the menu

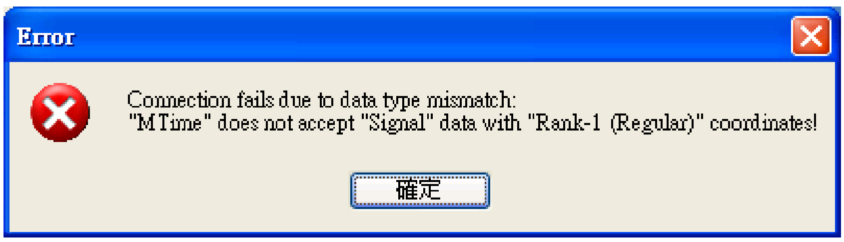


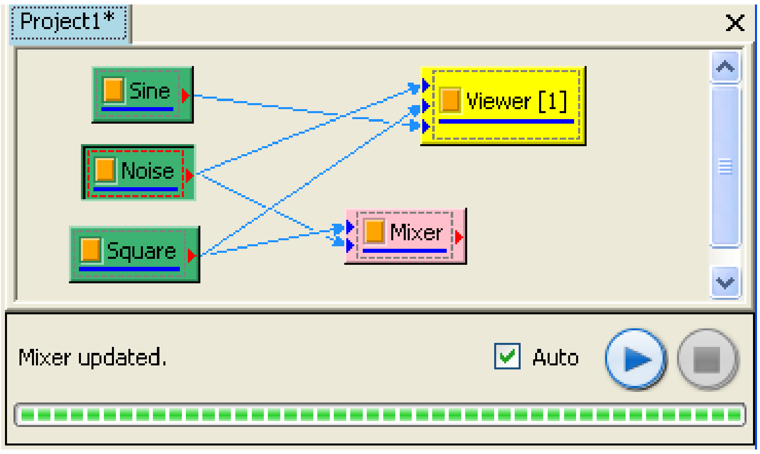
The advantage of the second method is that when you try to create the second SFO, all the available SFOs will be shown and the ones unavailable will be grayed out.



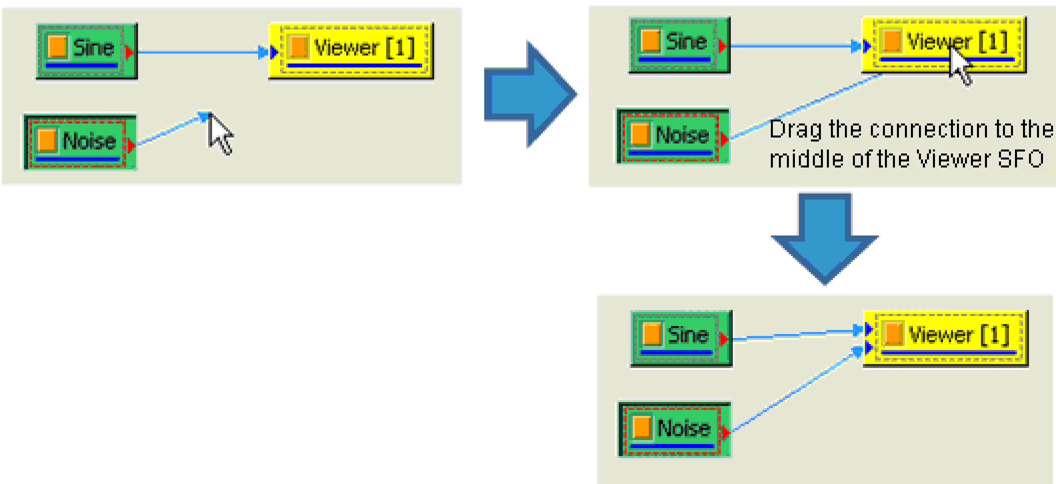
You can also create two SFOs and use the mouse to drag them together. In the example below a connection is established by dragging the red triangle of the Sine SFO to the blue triangle of the EEMD SFO.

The advantage of this method is that you can connect any SFOs at will without going through the menu to find a second SFO to connect to. This method gives you more freedom and control but it also leaves rooms for error. If you try to connect two SFOs of different input to output data then an error message will pop up not allowing you to connect them together (shown in the image below).





When trying to connect multiple SFO output to a single SFO input, the order you drag the connection from the output port to the input port will be the order it is shown on the input port. The first connection will be listed at the top followed by the second connection and so on.



2. To remove a connection between SFOs.

It is very easy to remove a connection that is connecting two SFOs together. Click on the connection arrow head which is currently connected to the input port of the second SFO and hold on the mouse to drag the arrow head back to the first SFO. By releasing the arrow head back into the first SFO, the connection between the two SFOs will be deleted.

3. Connection types

Different types of SFO will have different output connection types and they differ by their looks e.g. line thickness, dashed line and different colors.

Line Color	Definition
Blue	Real Signal
Purple	Complex Signal
Red	Spectra
Light Brown	Annotation Connection
Thin	Single-Channel Connection
Thick	Multi-Channel or Spectra Connection
Dashed	Unavailable Connection (no calculation)

4. Signal Flow Object warning

When a SFO does not receive any input data or part of the output data is missing, an 'exclamation mark' will appear on the Switch of the SFO to notify that there is a calculation mistake.

Chapter 3: Compute Signal Flow Object

3.1 Channel

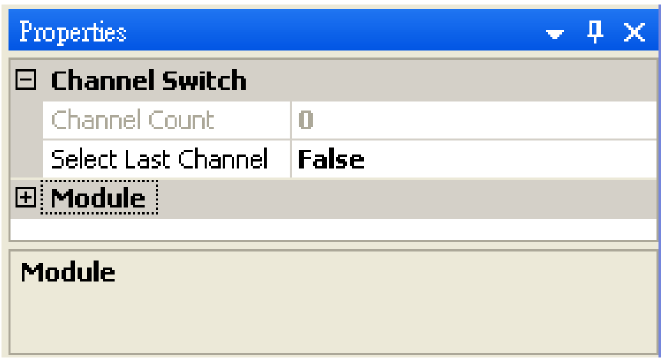
1. **Channel Switch:** Select a single-channel from a multi-channel source.
2. **Data Selection:** Select a time frame from a source data to be analyzed.
3. **Dup:** Duplicate a signal data.
4. **Fill NULL Value:** Use mathematical methods to fill any data that is missing (*NULL* value).
5. **Remove Channel:** Remove a single-channel from a multi-channel source.
6. **Replace Value:** Replace a particular value in the signal data.
7. **Resample:** Set a new sampling frequency value to a signal data.
8. **Time Shift:** Shift the graph along the x-axis (time).
9. **Data Merge:** Merge two signal data.
10. **Input Switch:** Accept all sorts of input signals, and one signal is chosen.

3.1.1 Channel Switch

Select a single-channel from a multi-channel source.

Properties

This module accepts input of Signal (which could be real number or complex number, single channel, Regular or indexed), numeric (which could be real number or complex number, single channel, Regular or indexed), and Audio (which could be real number or complex number, single channel, Regular). The output formats are real number, single channel, and Regular signal. The properties and settings of the Channel Switch are introduced below.

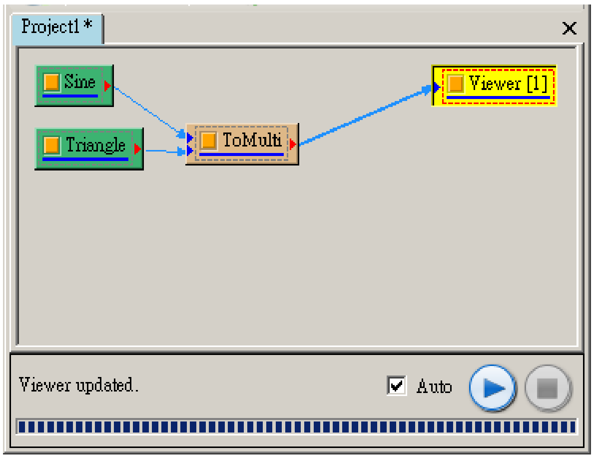


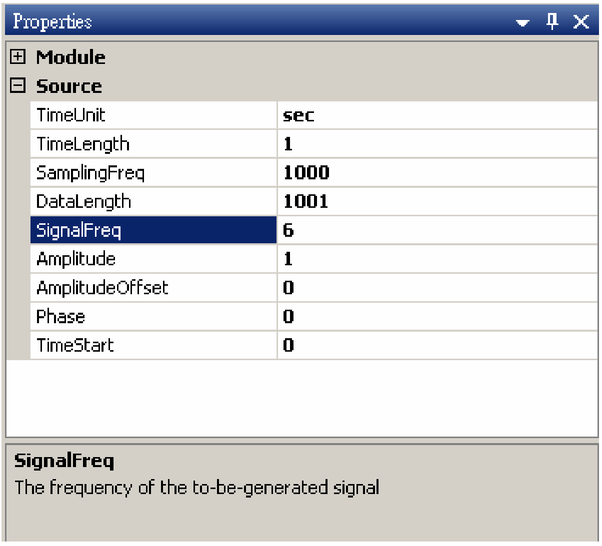
Property Name	Property Definition	Default Value
Channel Count	Shows the number of channel currently connected to the SFO.	Positive integer
Active Channel	Select the active channel.	Channel 1 (the 1^{st} channel)
Select Last Channel	If Select Last Channel is set as True, then the channel to be removed will always be the last channel.	False

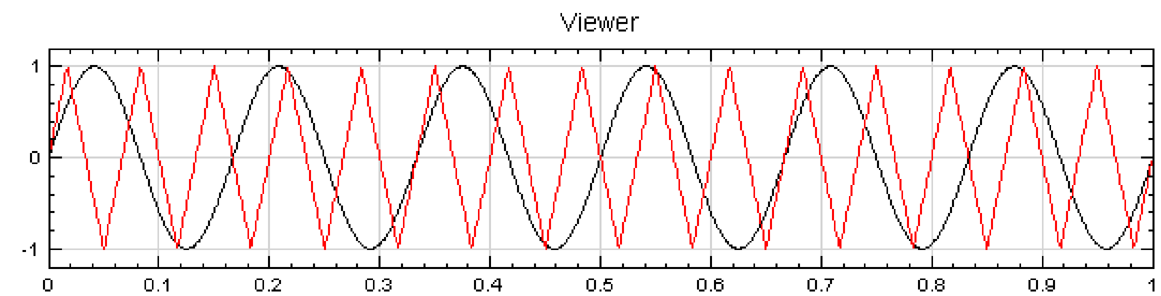
Example

Combine a sine wave data with a triangle wave data into multi-channel and use Channel Switch to select one of the signal waves to display.

1. Create a *Source→Sine Wave* and a *Source→Triangle Wave* and connect the two SFOs into a *Conversion→Merge to Multi-Channel* to create a multi-channel signal data.

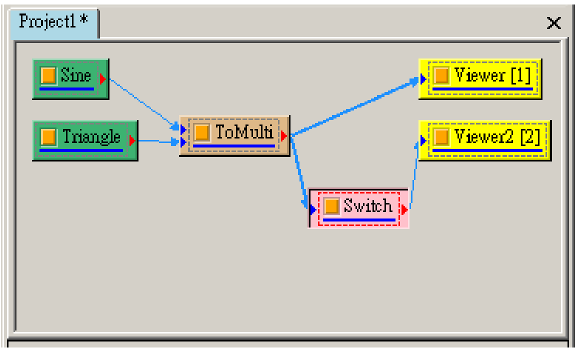


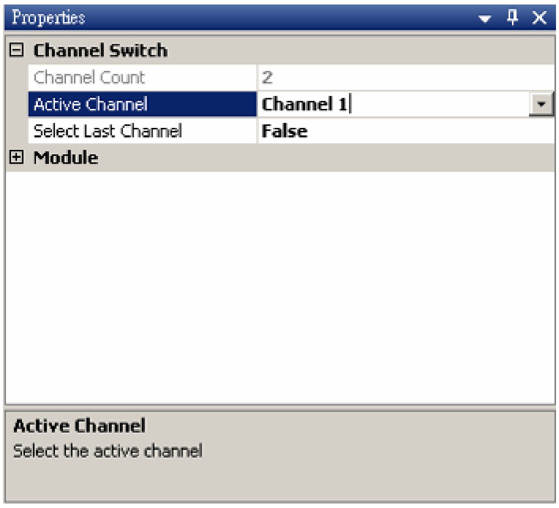


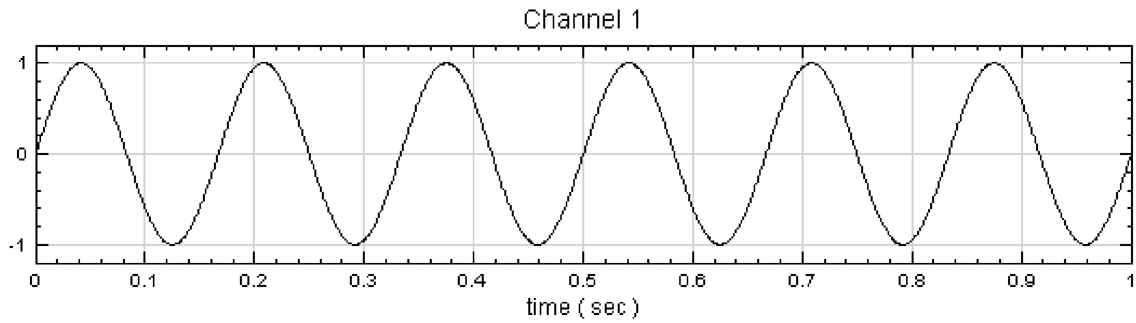


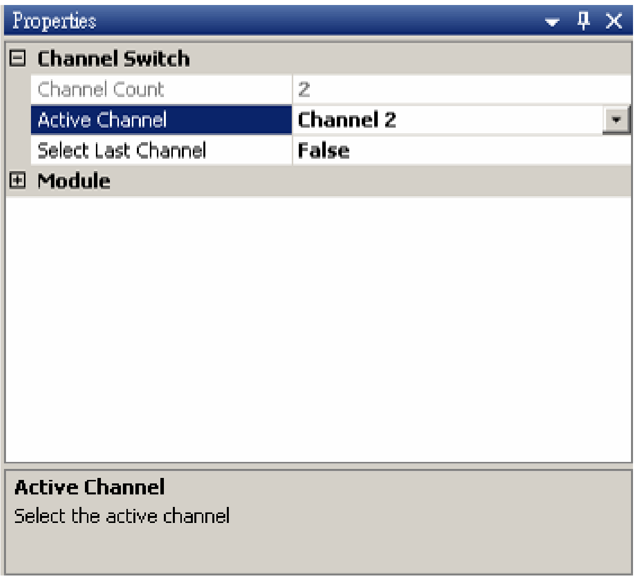
Change the *SamplingFreq* of both Sine and Triangle SFO to 1000, SignalFreq to 6 and 15 respectivley.

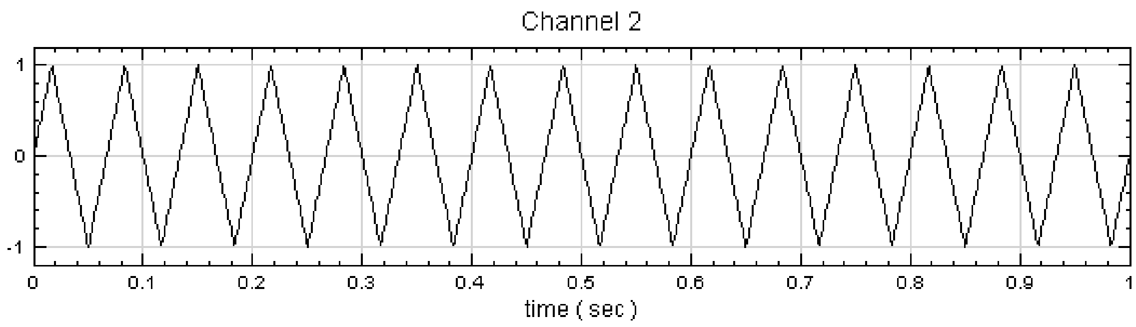
2. Output ToMulti SFO to a Channel Viewer SFO. In the Channel Switch SFO, you can change the *Properties/Active Channel* to Channel 1 or Channel 2 to read either the Sine wave signal or the Triangle wave signal.


Project1 *
Sine
Triangle
ToMulti
Switch
Viewer [1]
Viewer2 [2]


Properties
Channel Switch
Channel Count
2
Active Channel
Channel 1
Select Last Channel
False
Module
Active Channel
Select the active channel


Channel 1
1
0
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time (sec)


Properties
Channel Switch
Channel Count
2
Active Channel
Channel 2
Select Last Channel
False
Module
Active Channel
Select the active channel



Related Functions

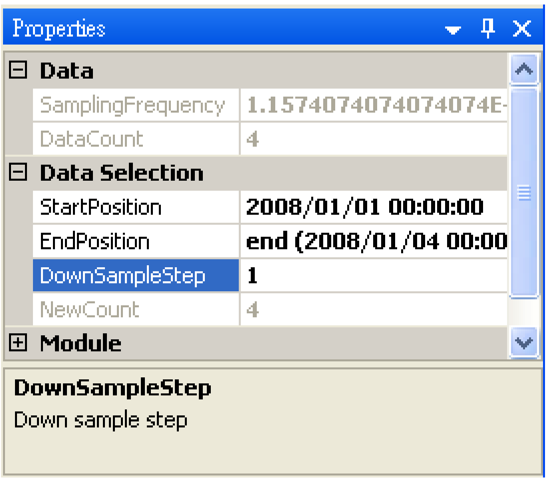
Merge to Multi-channel, Viewer, Source.

3.1.2 Data Selection

Select a time frame from a source data to be analyzed.

Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular) and Audio (which could be real number or complex number, single channel or multi-channel, Regular). Enter the selected range of the signal by defining the *Properties/StartPosition* and *Properties/EndPosition* (time unit).



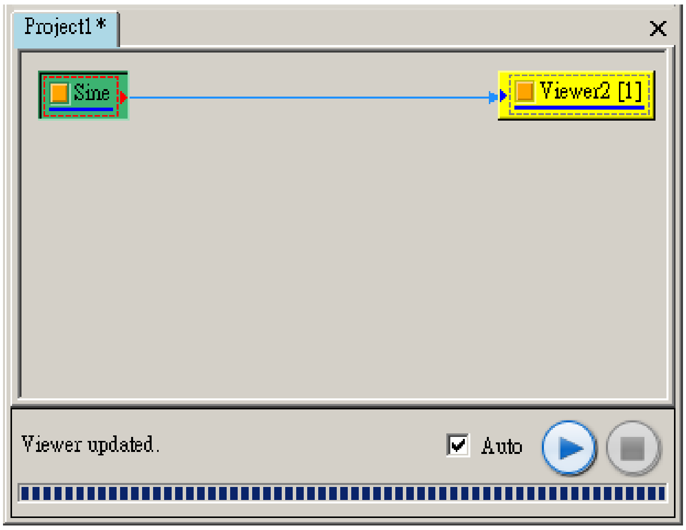
Data Property Name	Property Definition
SamplingFrequency	Displays the sampling frequency of the input data
Data Count	Displays the sampling count of the input data.

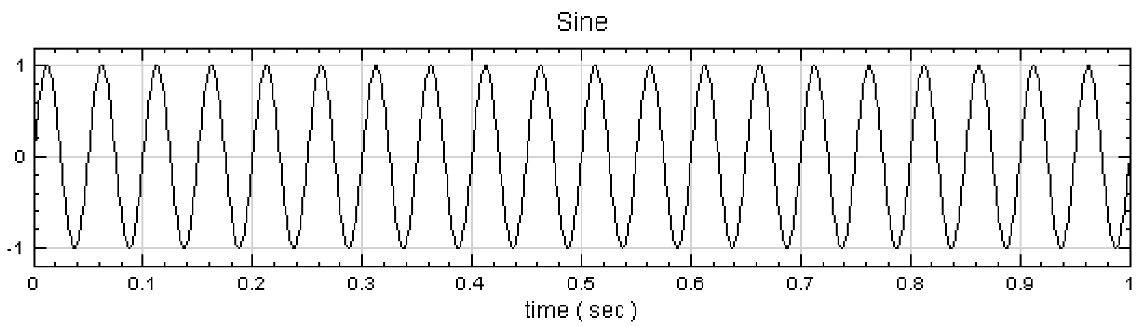
Data Selection Property Name	Property Definition	Default Value
StartPosition	Enter the value of the start position of the input data.	The original start time for the input data.

EndPosition	Enter the value of the end position of the input data.	The original end time for the input data.

Example

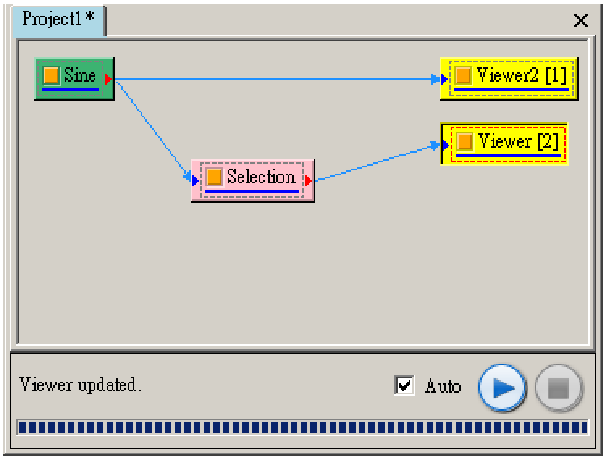
1. Create *Source→Sine Wave* and connect it to *Viewer→Channel Viewer*.

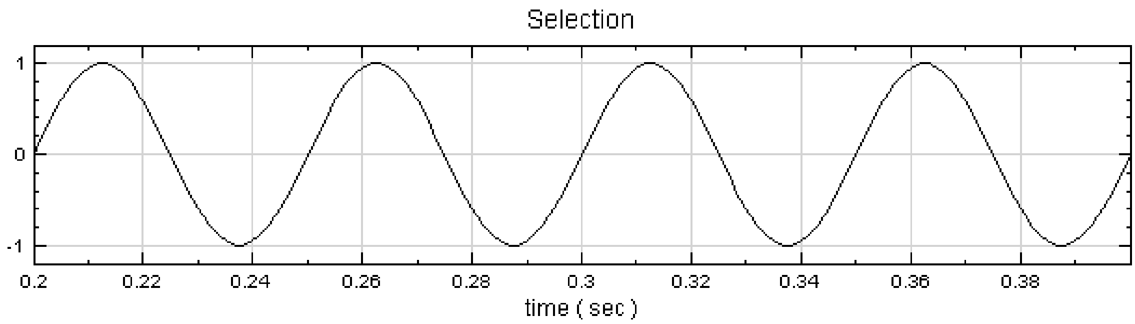




In this Sine Wave SFO the SamplingFreq is 1000 and the SignalFreq is 20.

2. In DataSelection SFO set the *Start Position* to t=0.2 and *End Position* to t=0.4 and the new graph will be show the range between t=0.2 to t=0.4 (original range *Timelength* is between [0,1] so the new range has to be within the original range).





Related Functions

Channel Switch, Viewer, Source.

3.1.3 Dup

Duplicate a signal data.

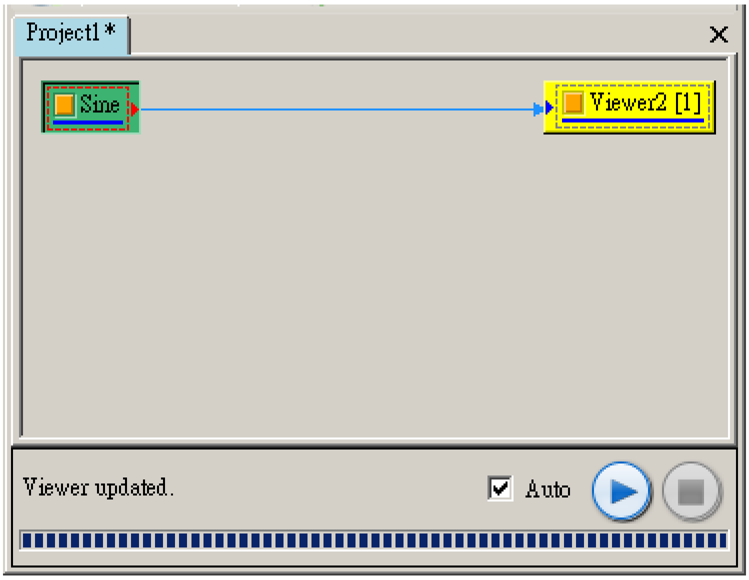
Properties

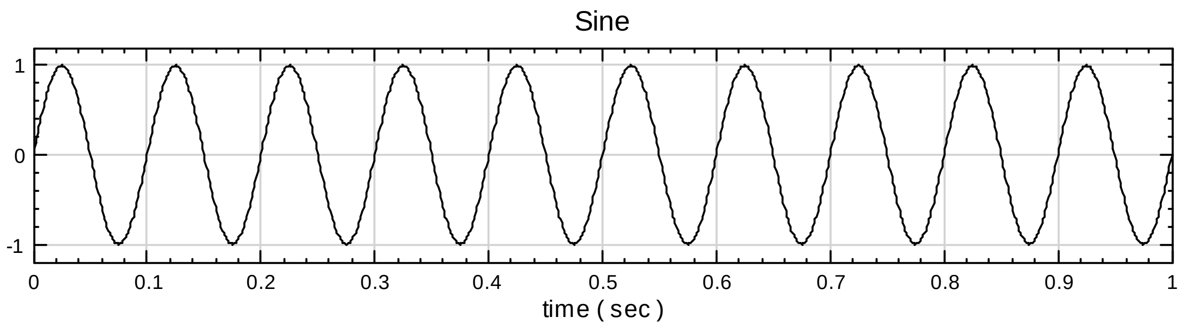
Dup can take input from any type of data input and it does not require to configure anything.

Example

To duplicate a Sine Wave SFO using Dup.

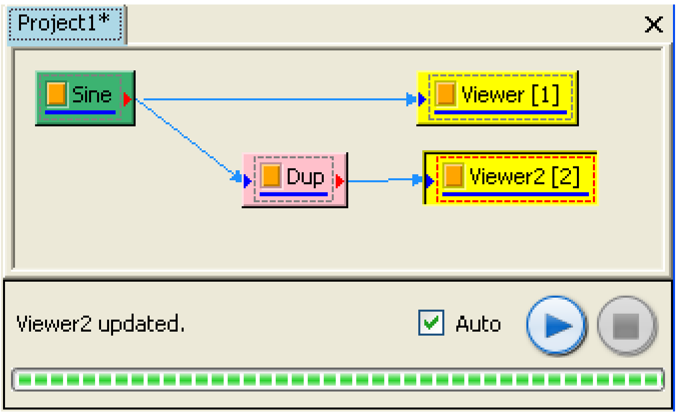
1. Firstly create *Source*→*Sine Wave* can connect it to *Viewer*→*Channel Viewer*.

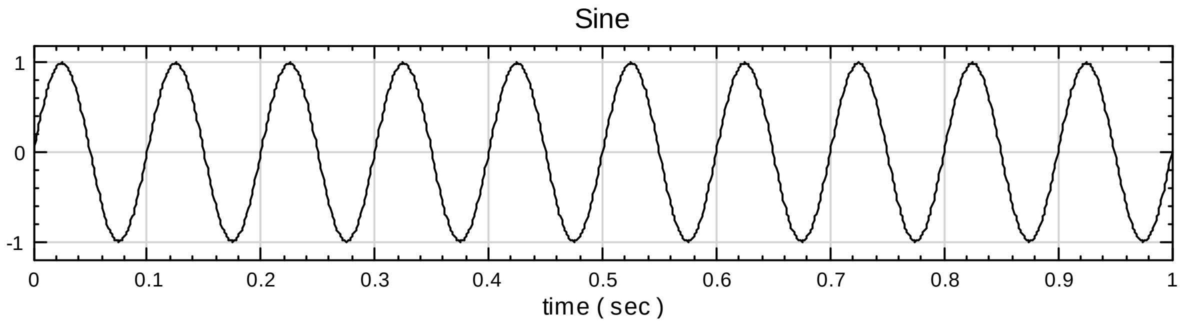




In this Sine Wave SFO the SamplingFreq is 1000 and the SignalFreq is 10.

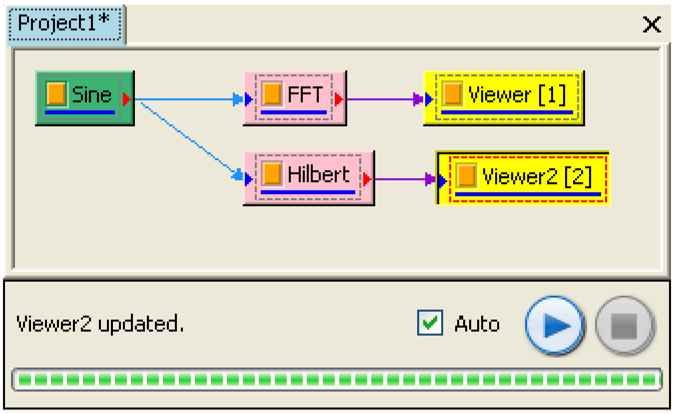
2. After connecting Sine Wave SFO to a DUP SFO, you can see that both signal data looks the same through the Channel Viewer SFOs.



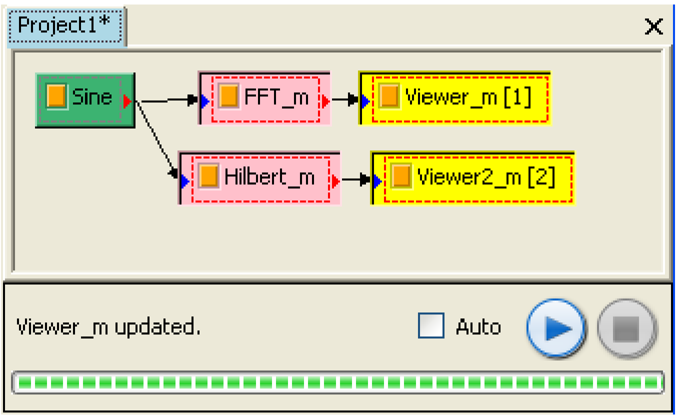


Below is an example of saving a DUP scenario into a Macro.

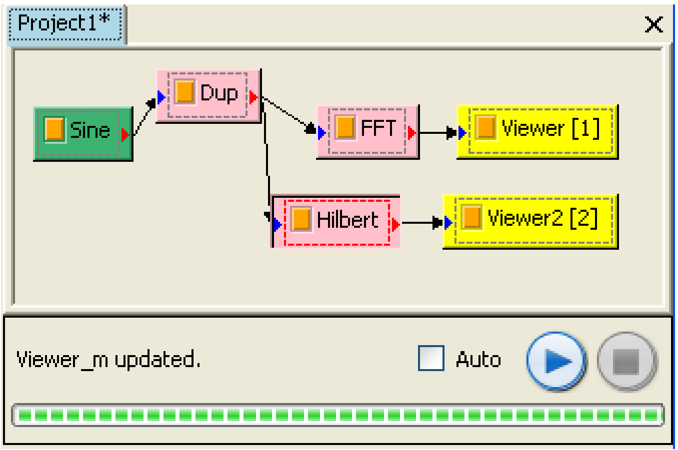
Connect *Source* → *Sine* to *Compute* → *Transform* → *Fourier Transform* and *Compute* → *Transform* → *Hilbert Transform* and view each result through *Viewer* → *Channel Viewer*.



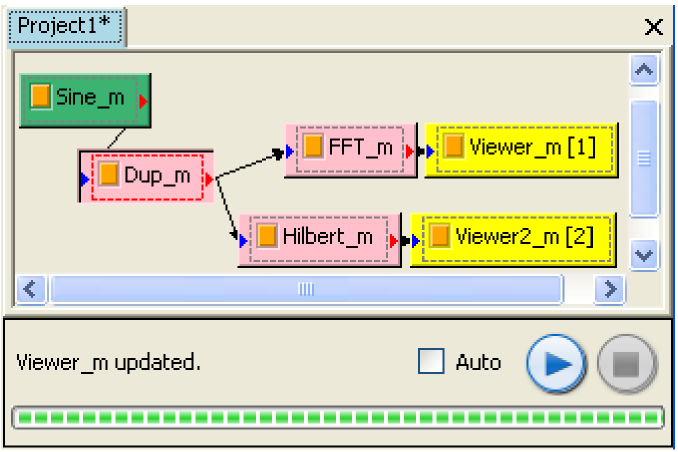
Save this Signal Flow Diagram as a macro file. Now load the macro file into a clean project and the result will be as shown:



Now you want to change the Source SFO to another source. After deleting the old Source SFO, create a new Source SFO and connect it back to the other SFOs. If the macro contains numerous numbers of SFOs, reconnecting a new Source SFO back to the network relationship can be tedious. So the easy way is to create a *Compute→Channel→Dup* in place of the current position of the Source SFO and connect the Source SFO to the Dup SFO.



Save the macro with dup and load the macro file.



Whenever a new macro is loaded, just connect the Source SFO to the Dup SFO without having to reconnect the Source SFO to the rest of the network relationship since the Dup SFO is already connected to the rest of the network relationship.

Related Functions

Macro.

3.1.4 Fill NULL Value

Use mathematical method to fill any data that is missing with the *NULL* value.

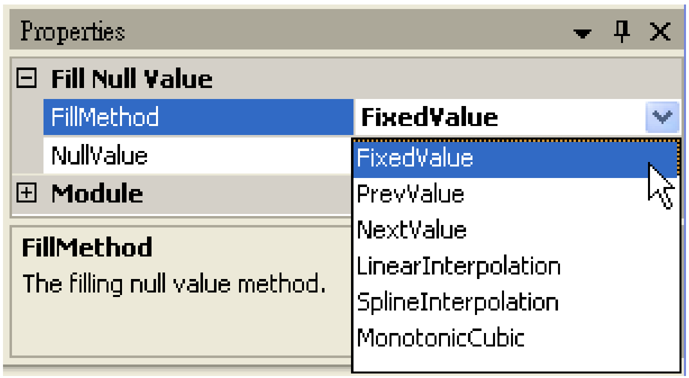
Introduction

To fill in the data signal $X = \{x_0, x_1 ... x_{N-1}\}$ which contains *NaN* (Not A Number) or *NULL*.

Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular or Indexed) and Audio (which could be real number, single channel or multi-channel, Regular).

In the *Properties/Fill NULL Value/Fillmethod* there are 6 methods to fill in the missing values.



Property Name	Property Definition	Default Value
FillMethod	There are the FixedValue, PrevValue, NextValue, LinearInterpolation, SplineInterpolation and MonotonicCubic methods to fill the *NULL* value.	SplineInterpolation

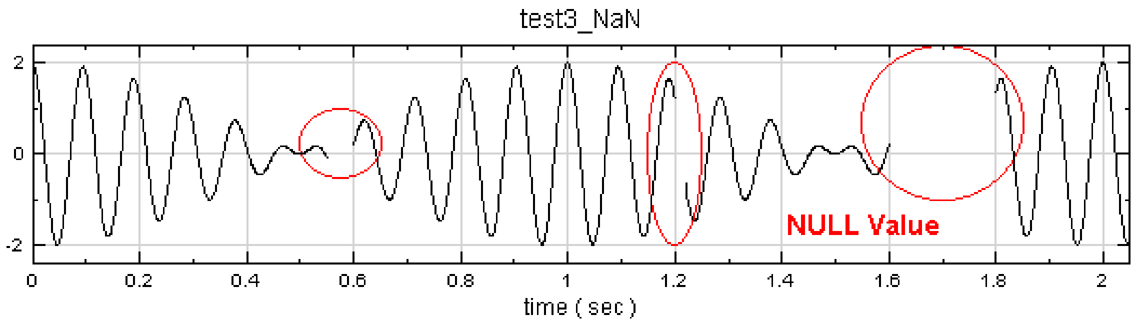
Variable Option	Property Definition
FixedValue	A new FixedValue variable option will appear in the *Properties Window*. Enter a value to replace all the *NULL* values to the value entered.
PrevValue	The *NULL* value will be replaced with the previous value in

	the signal.
NextValue	The *NULL* value will be replaced with the next available value in the signal.
LinearInterpolation	Using Linear Interpolation to calculation the value of the *NULL*.
SplineInterpolation	Using Spline Interpolation to calculation the value of the *NULL*.
MonotonicCubic	Monotone cubic interpolation is a type of cubic interpolation that preserves monotonicity of the data set being interpolated. MonoticCubic method is better than SplineInterpolation method when the slope of the signal is large e.g. Square wave.

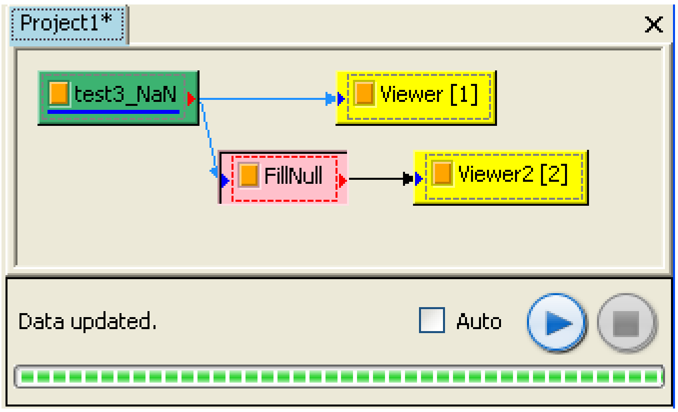
Example

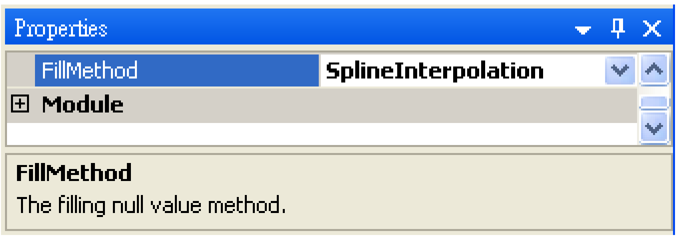
To fill in the missing values using Fill NULL Value SFO.

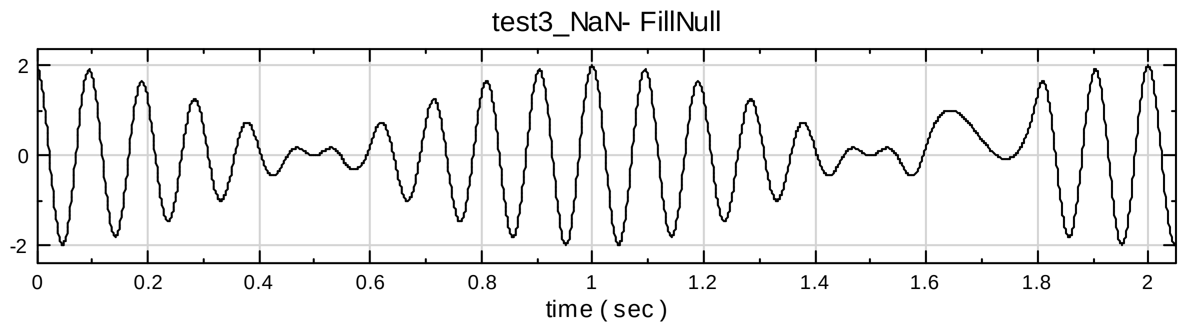
1. Open demo53 in the directory C:\Program Files\DynaDx\DataDemon\data. From the graph in the *Visualization Window*, you can clearly see the missing values on the graph.



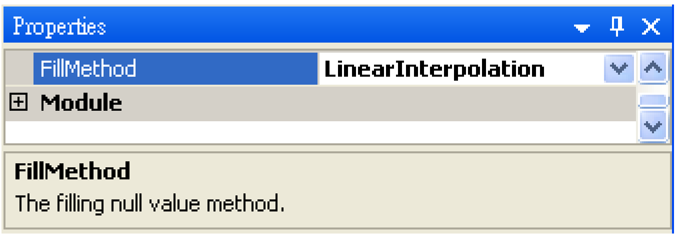
2. Connect the source signal data to *Compute→Channel→Fill NULL Value* and select *Spline Interpolation* in the *Properties/FillMethod* of the Fill NULL Value SFO.

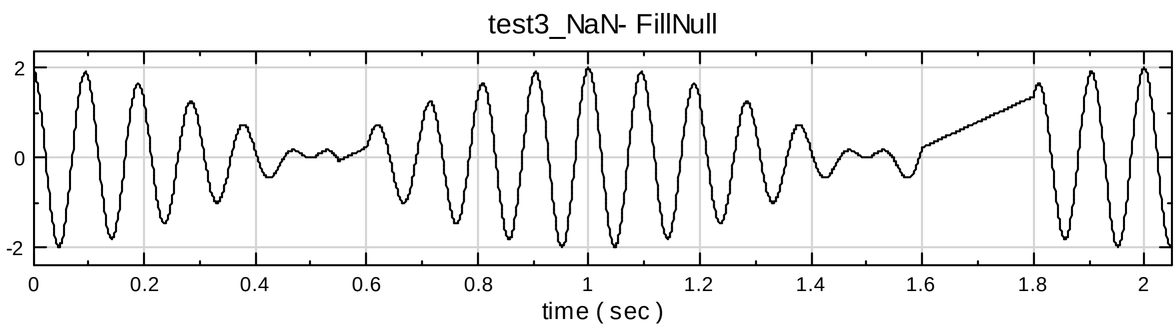




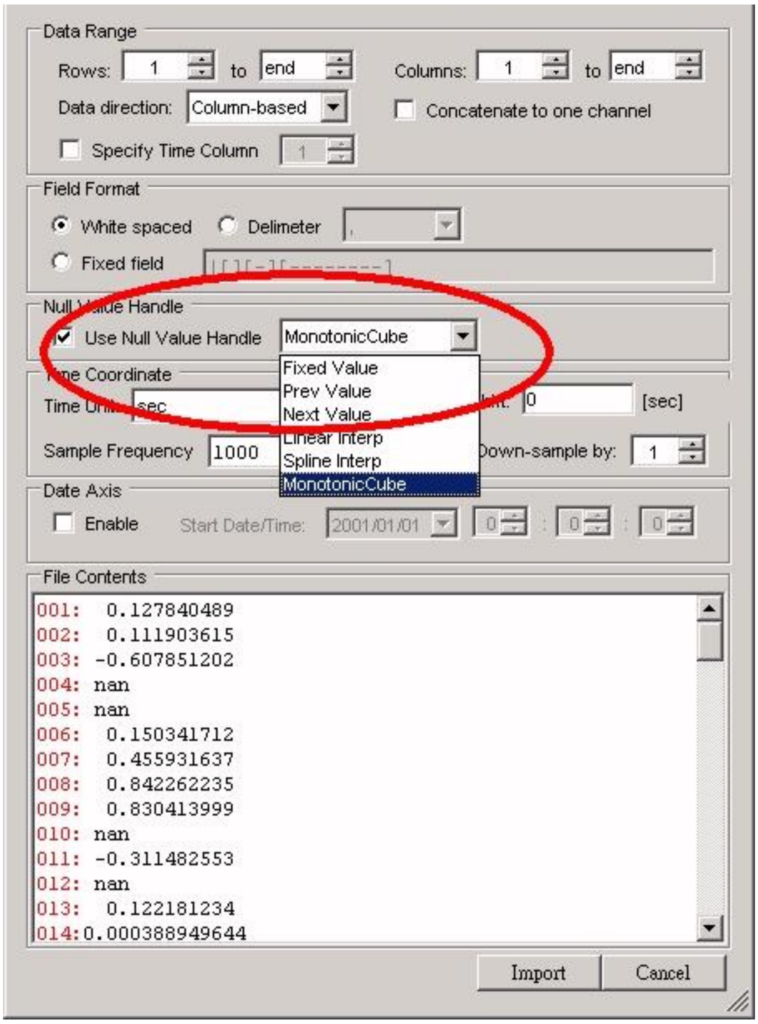


3. Select *Linear Interp* method instead and the way the values are filled in will be considerably different.

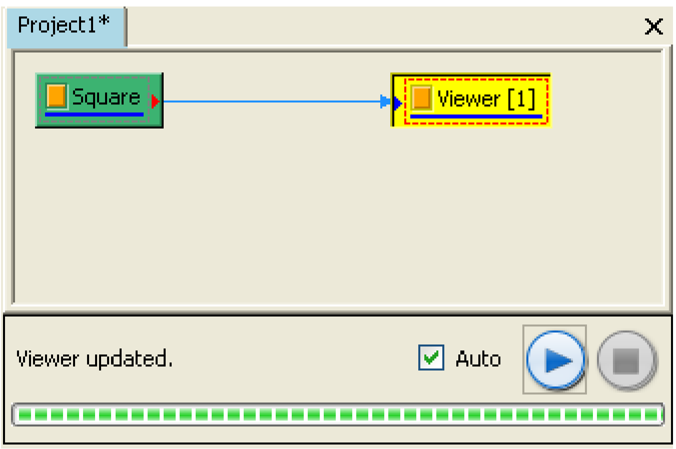


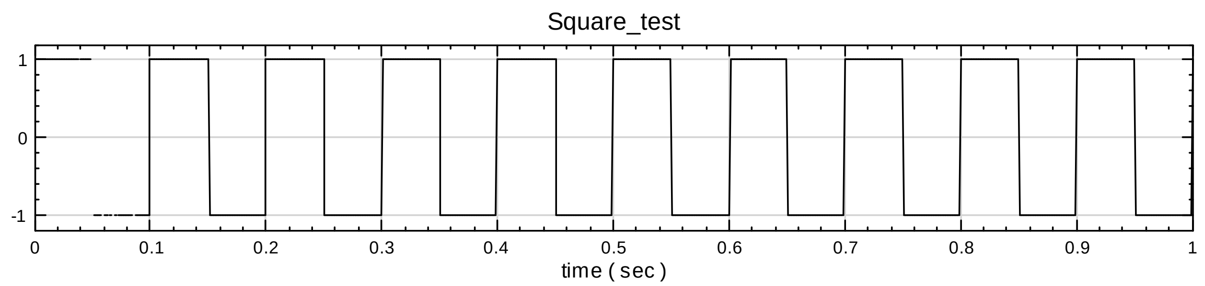


4. In the Data Importer there is also an option to fill in the missing value but this feature is different from the Fill NULL Value SFO.

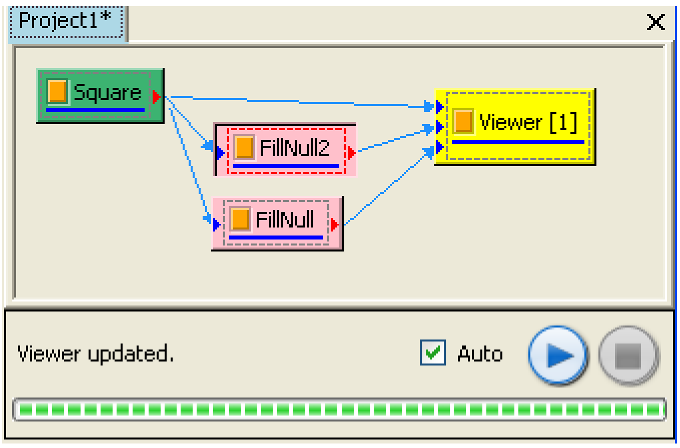


Import a data which has missing values, and intentionally unchecked the box of *Use NULL Value Handle* in Data Importer. The imported value and graph is shown in the image below.

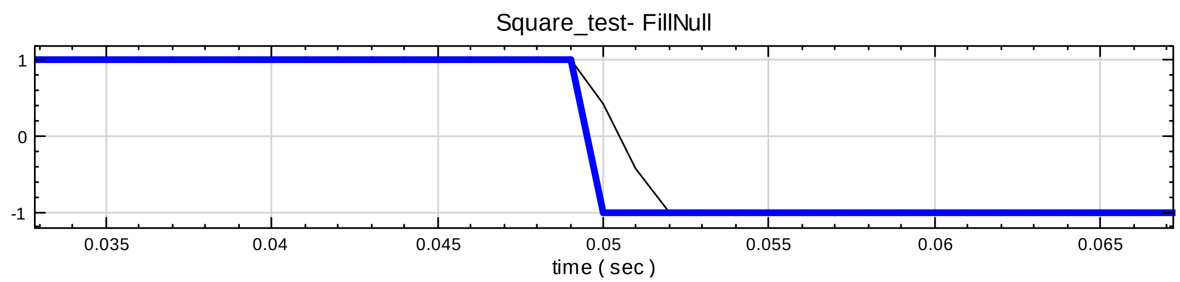




Now create two Fill NULL Value SFOs to connect from the imported source signal data. The first SFO will use the Spline Interpolation fill in method and the second SFO use Monotonic Interpolation fill in method.



From the results shown in the Viewer, there is an obvious difference between the Spline Interpolation method (thin dark line) and the Monotonic Interpolation method (thick blue line).



Related Functions

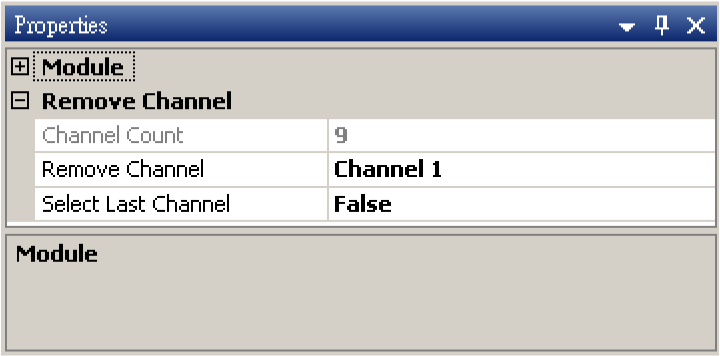
Data Importer, Resampling, Data Importer.

3.1.5 Remove Channel

Remove a single-channel from a multi-channel source.

Properties

This module accepts input of Signal (which could be real number or complex number, multi-channel, Regular or Indexed) and Audio (which could be real number or complex number, multi-channel, Regular).

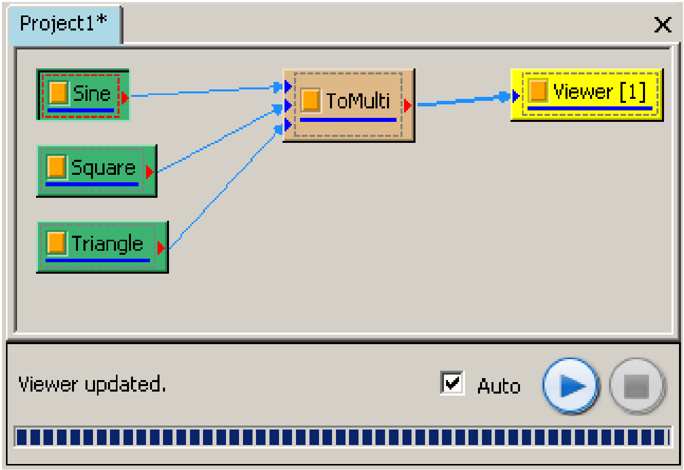


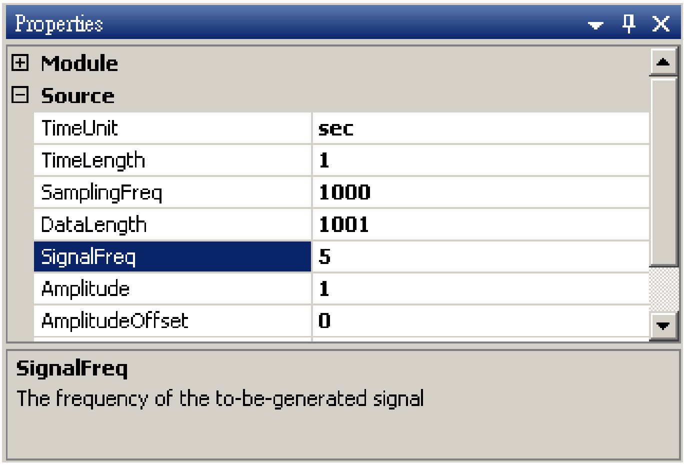
Property Name	Property Definition	Default Value
Channel Count	Displays the number of channels.	NONE
Remove Channel	Select the channel to be removed.	Channel 1
Select Last Channel	If Select Last Channel is set as True, then the channel to be removed will always be the last channel.	False

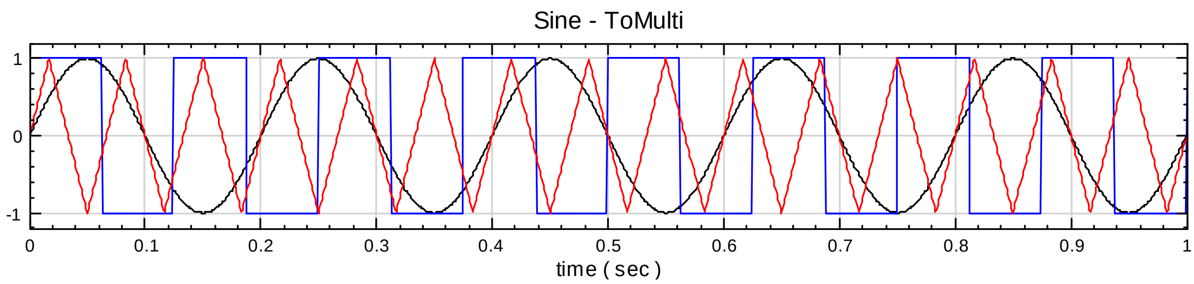
Example

Combine a sine wave and a triangle wave and a square wave together, connect it to a Remove Channel SFO and to remove the sine wave.

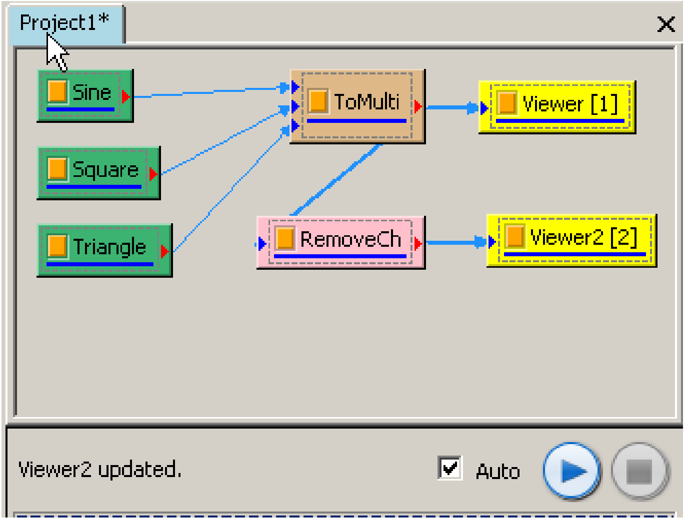
1. *Create Source→Sine Wave*, *Square Wave* and *Triangle Wave* to connect them all to a *Conversion→Merge to Multi-Channel* to make the three waves into a Multi-channel signal data. Set the SamplingFreq as 1000 and set the Sine SFO's *SignalFreq* as 5, Square SFO's *SignalFreq* as 8 and Triangle SFO's *SignalFreq* as 15 to observe the different waves on the graph.

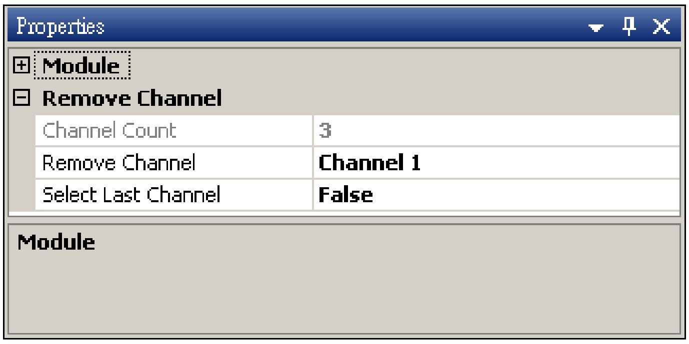


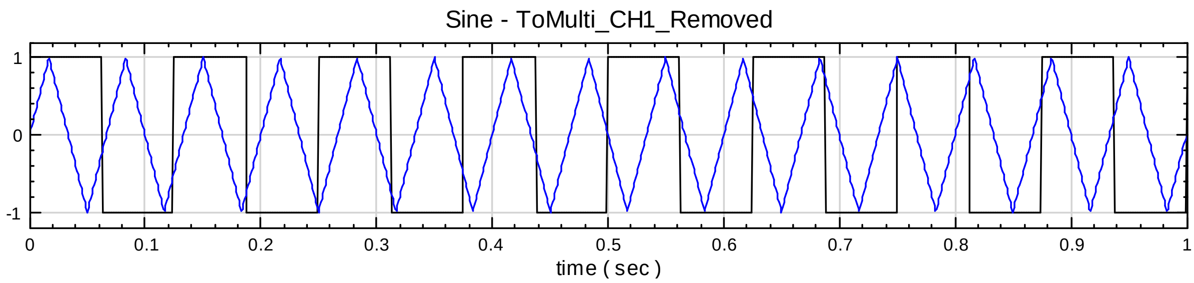




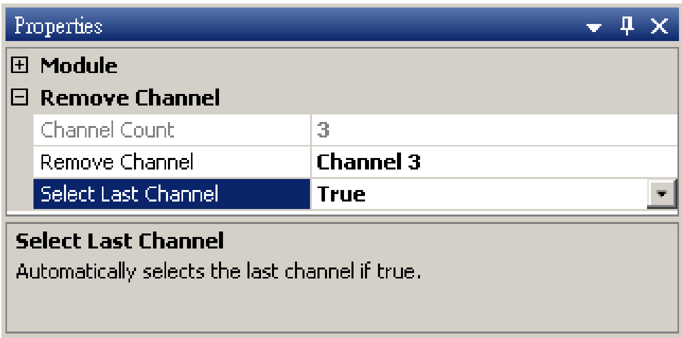
2. Connect Merge to Multi-Channel SFO to *Compute→Channel→RemoveChannel* and select *Properties/Remove Channel* as Channel 1 (sine wave).

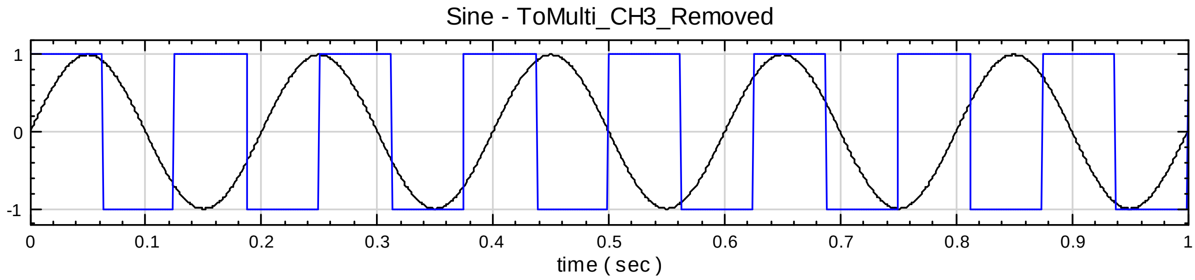






3. Now set the *Properties/Select Last Channel* as True and you'll see that *Remove Channel* will automatically change to Channel 3 and the triangle wave signal (Channel 3) will be removed.





Important:

Channel Switch and *Remove Channel* is completely the opposite. *Channel Switch* preserves a single selected channel from a multi-channel signal data and *Remove Channel* removes the single selected channel from a multi-channel signal data.

Related Functions

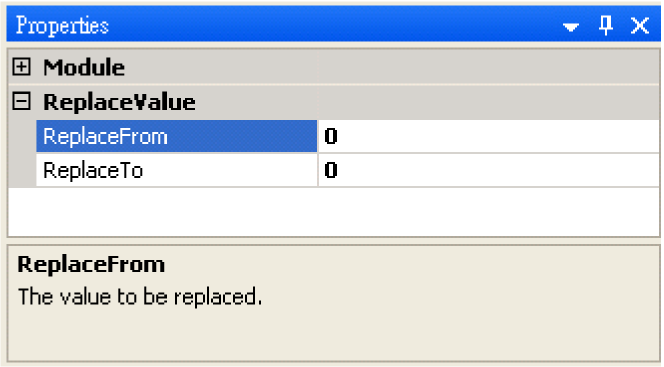
Merge to Multi-Channel, Channel Switch.

3.1.6 ReplaceValue

Replace a particular value in the signal data.

Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular).

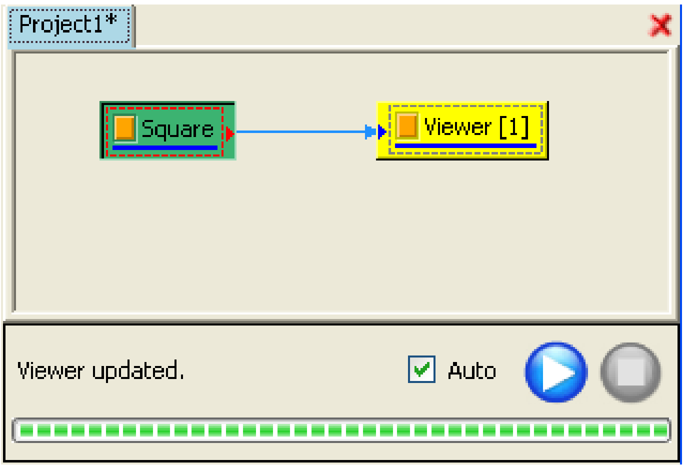


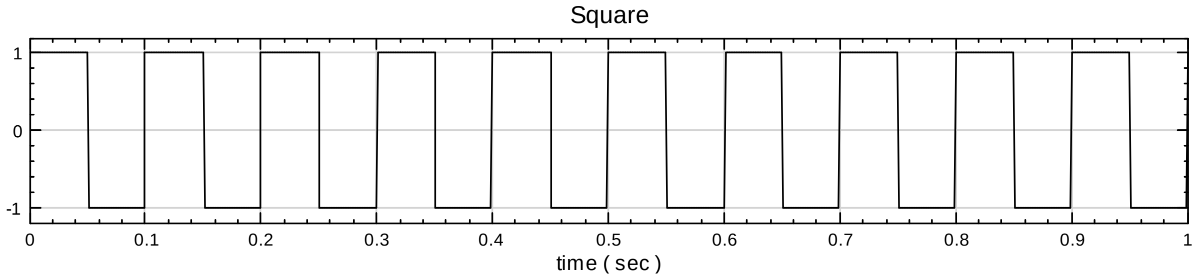
Property Name	Property Definition	Default Value
ReplaceFrom	Set the value to be replaced.	0
ReplaceTo	Replace the set value with this new value.	0

Example

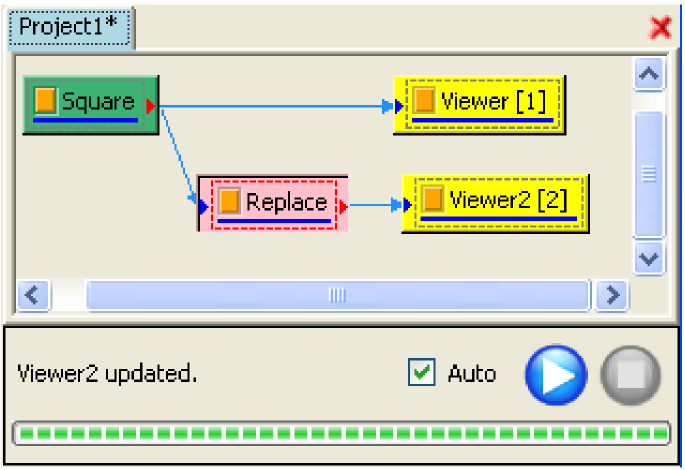
Change the maximum value of the square wave to another number.

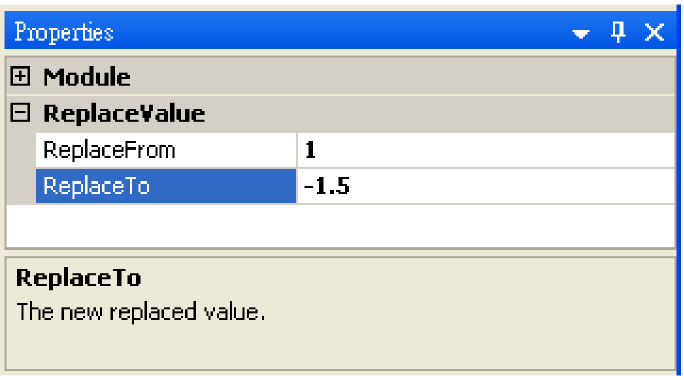
1. Create a *Source Square→Wave* and connect it to a *Viewer Channel→Viewer*.

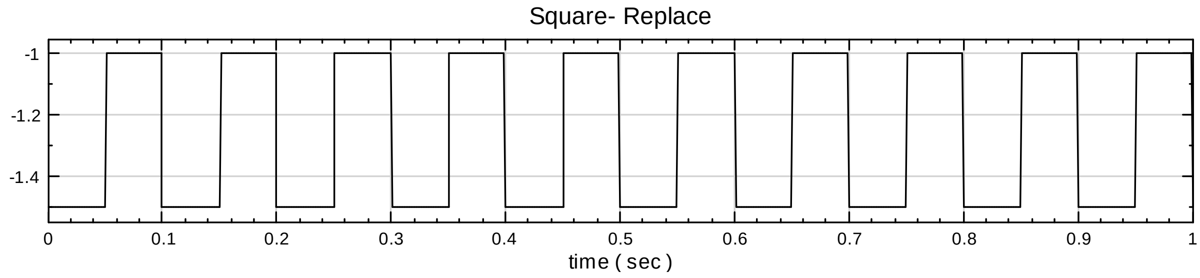




2. Now connect the Sine SFO to *Compute→Channel→ReplaceValue* and set the ReplaceValue SFO's *Properties/ReplaceValue/ReplaceFrom* to -1.5. Now all the values of the square wave which was originally 1 will become -1.5.







Important:

You can only replace one value at a time. If you want to replace multiple values then several Replace Value SFO will have to be created.

Related Functions

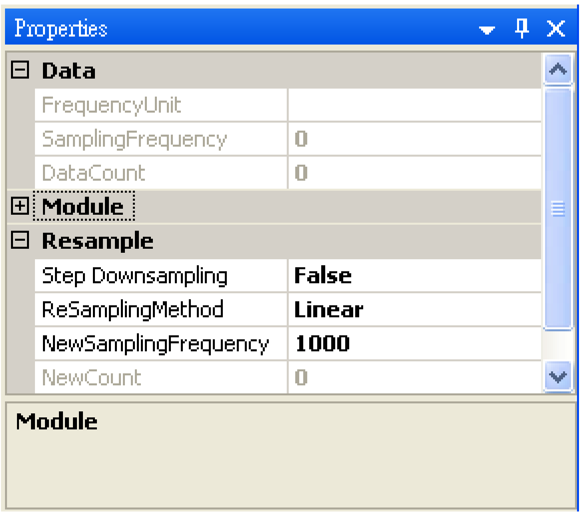
None

3.1.7 Resampling

You can set a new sampling frequency value to a signal data.

Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular) and Audio (which could be real number or complex number, single channel or multi-channel, Regular).



Resample Property Name	Property Definition	Default
NewSamplingFrequency	Set the new sampling frequency value.	1000Hz
ReSamplingMethod	Set the sampling frequency method; nearest, linear, spline and mono tonic cubic.	Linear
NewCount	Display the new sampling count of the data output.	None

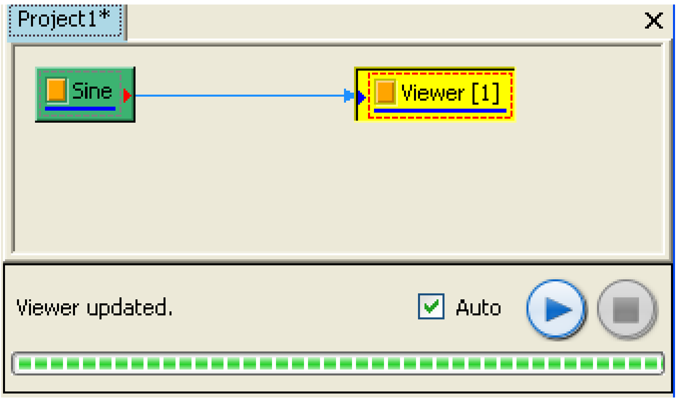
Variable Option	Property Definition
NextValue Nearest	The *NULL* value will be replaced with the next available value in the signal.
LinearInterpolation	Using Linear Interpolation to calculation the value of the *NULL*.

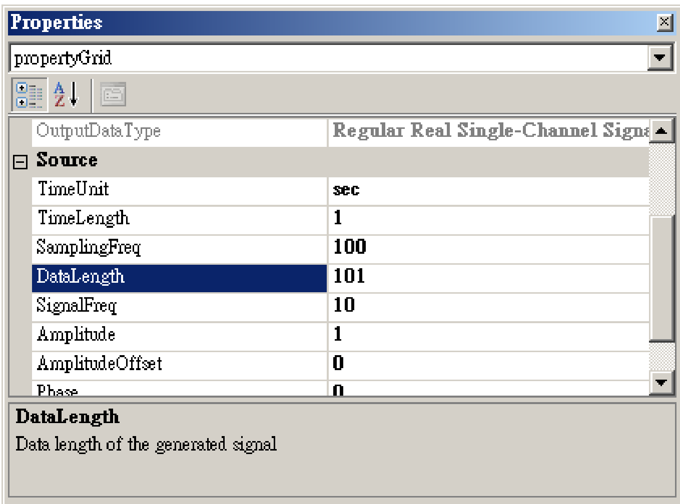
SplineInterpolation	Using Spline Interpolation to calculation the value of the *NULL*.
MonotonicCubic	Monotone cubic interpolation is a type of cubic interpolation that preserves monotonicity of the data set being interpolated. MonoticCubic method is better than SplineInterpolation method when the slope of the signal is large e.g. Square wave.

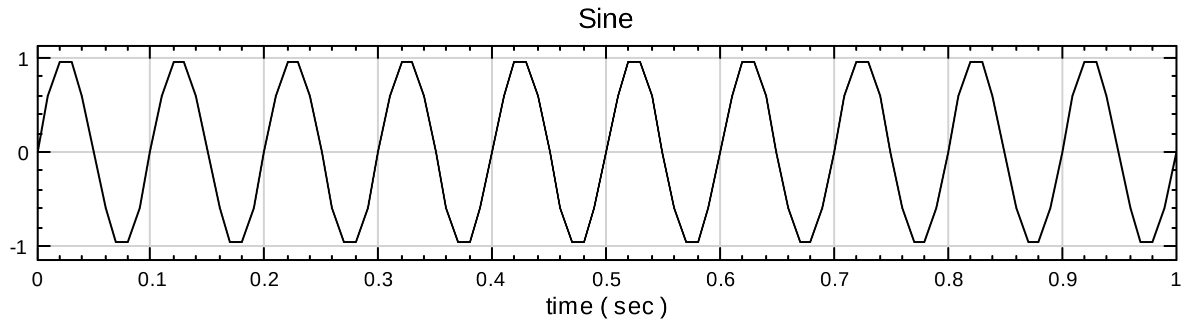
Example

Create a sine wave SFO and apply Resampling SFO to it.

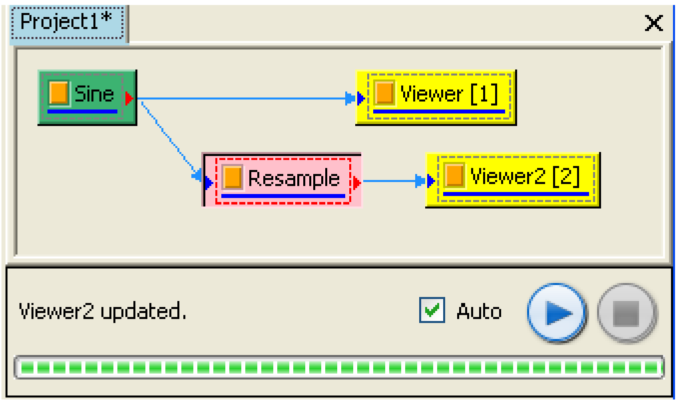
1. Create *Source→Sine Wave* and edit the *Properties/Source/SamplingFreq* to 100 and *Properties/Source/DataLength* to 101. Connect the Sine Wave SFO to *Viewer→Channel View* to see the graph. You can clearly see from the graph that the wave signal is not as smooth anymore.

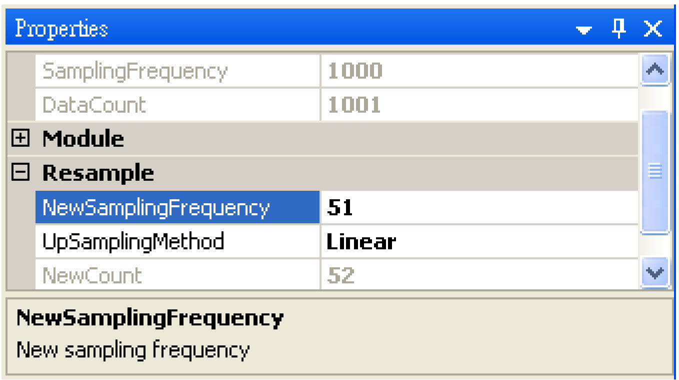


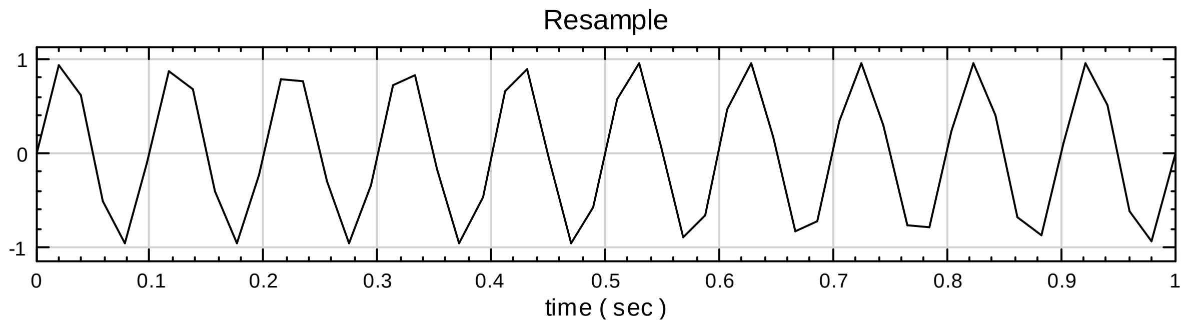




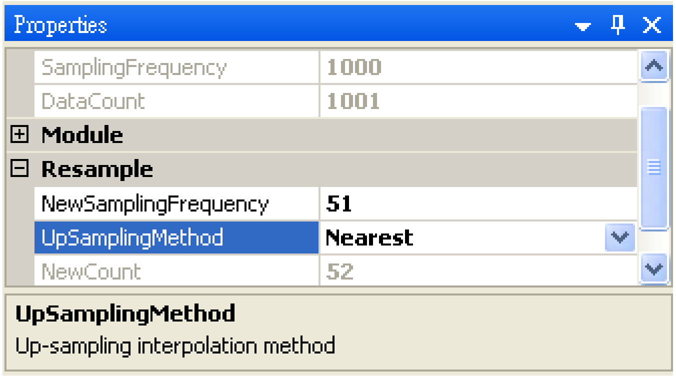
2. Connect the Sine wave SFO to *Compute→Channel→Resampling* and edit the value of *Properties/Resample/NewSamplingFrequency* to 51 and *Properties/Resample/UpsamplingMethod* to Linear to compare the difference between the two.

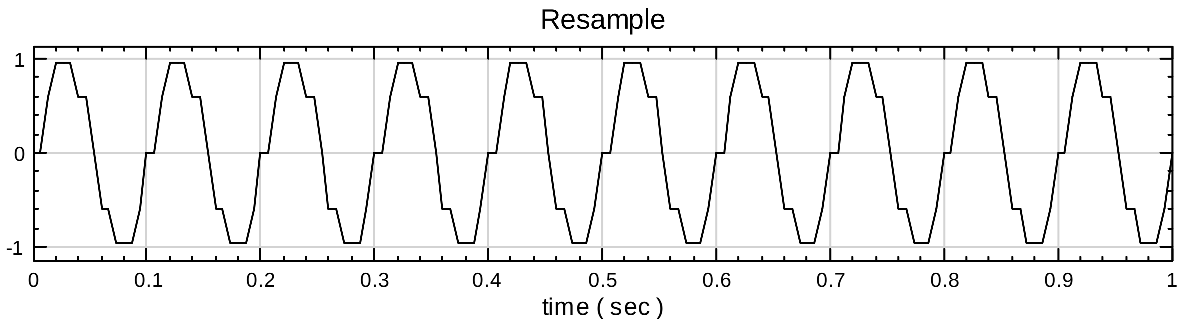




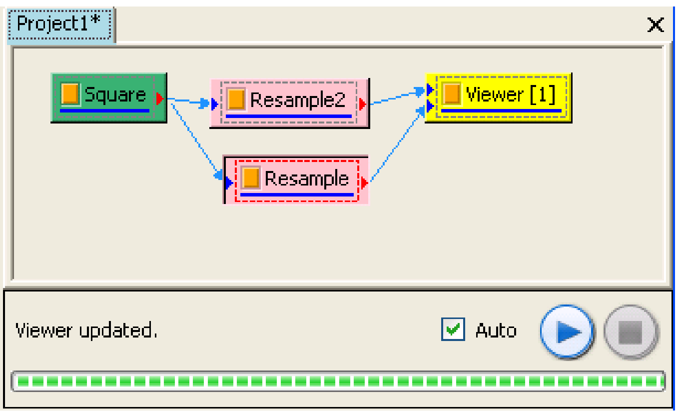


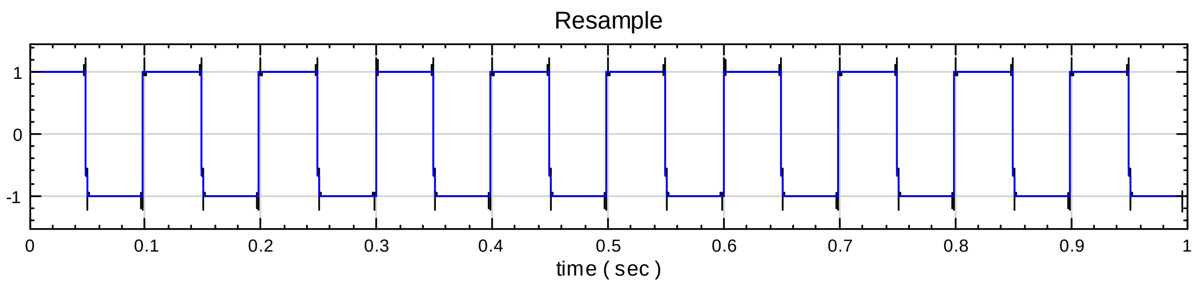
3. Now try with changing the *UpsamplingMethod* to *Nearest*.





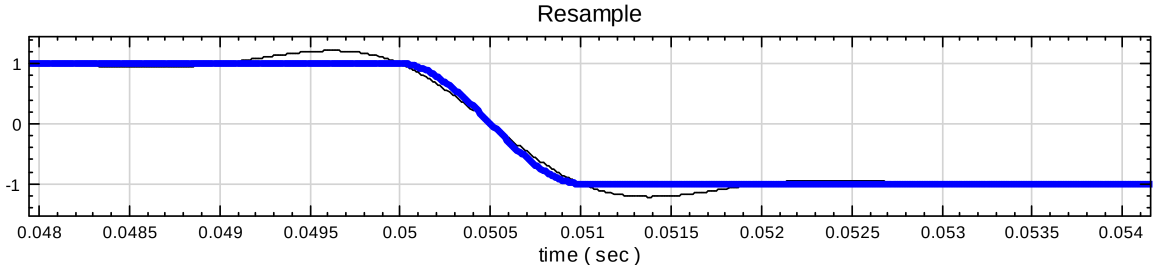
4. Now create a Square wave SFO and connect it to two Resampling SFOs and set one Resampling SFO's *NewSamplingFrequency* to 100000 and *UpSamplingMethod* to *Spline* and the other Resampling SFO's *NewSamplingFrequency* to 100000 and *UpSamplingMethod* to *MonotonicCubic* and connect both Resampling SFOs to the same Channel Viewer SFO.





Notice the slight different around the corners of both wave signals. Now lets

zoom into the graph for a closer look.



Overshooting. The thin black line is created through the Spline method and the thick blue line is created through the MonotonicCubic method. From the graph you can observe that Spline method has a tendency of overshooting whilst MonotonicCubic method has no such problem.

Related Functions

Filling NULL Value

Reference

Numerical Recipes 3rd Edition: The Art of Scientific Computing by William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery

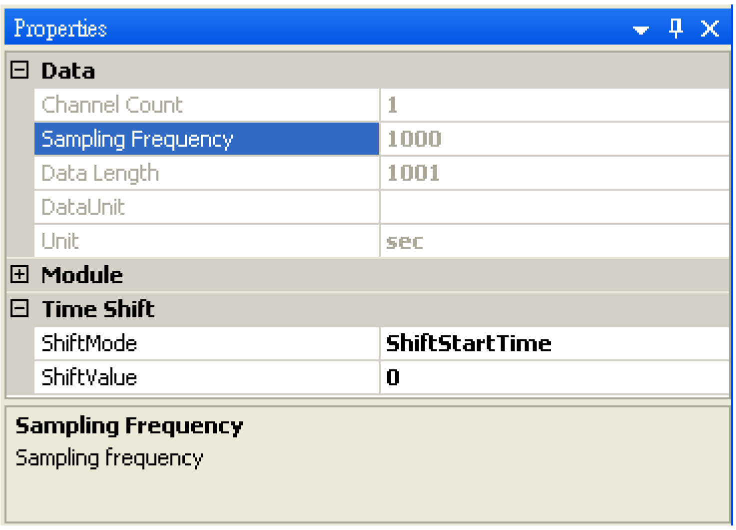
http://en.wikipedia.org/wiki/Monotone_cubic_interpolation

3.1.8 Time Shift

Shift the graph along the x-axis (time).

Properties

This module accepts input of Signal (which could be real number or complex, single channel or multi-channel, Regular) and Audio (which could be real number or complex, single channel or multi-channel, Regular).



Property Name	Property Definition
Channel Count	Displays the number of channels connected to the SFO.
Sampling Frequency	Displays the sampling frequency of the SFO.
Data Length	Displays the data length of the SFO.
Data Unit	Displays the data unit of the SFO.
Unit	Displays the unit of the SFO.

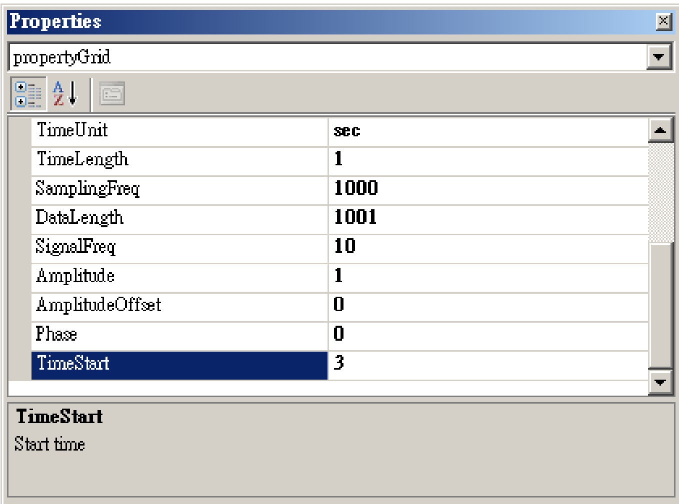
Variable Option	Property Definition	Default Value

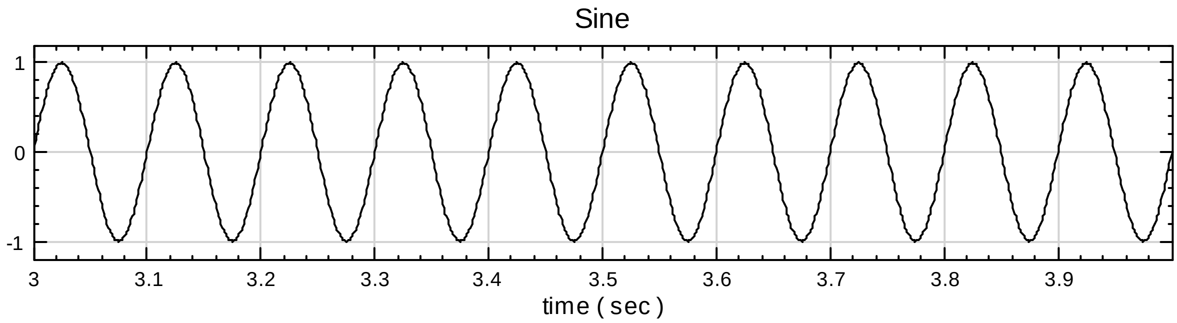
ShiftMode	Select the type of shift method to apply to the graph.	ShiftStartTime
ShiftStartTime	Shift Value. Shift the start time of the graph to the entered value (the time shift will either add to or minus from the original start time).	ShiftValue = 0
SetStartTime	Start Value. Set the start time of the graph to the entered value.	StartValue = 0
SetStartDate	Start Date, Start Time. Set the start date and the start time of the graph to the entered value.	StartDate = 2000/1/1 StartTime = 00:00:00

Example

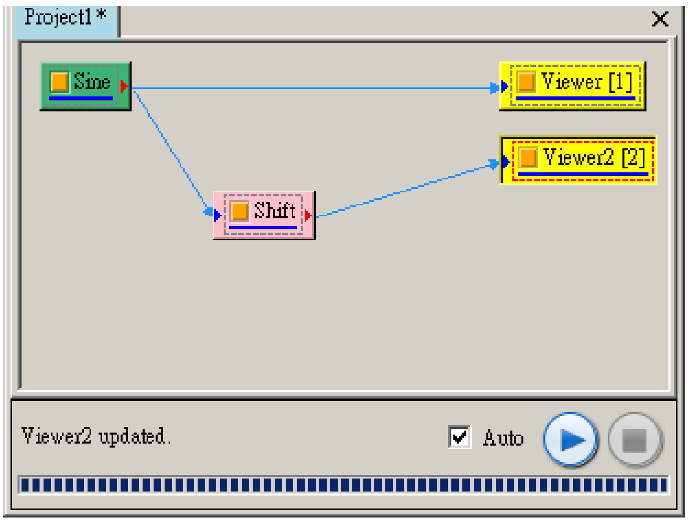
Create a sine wave SFO and shift its time value.

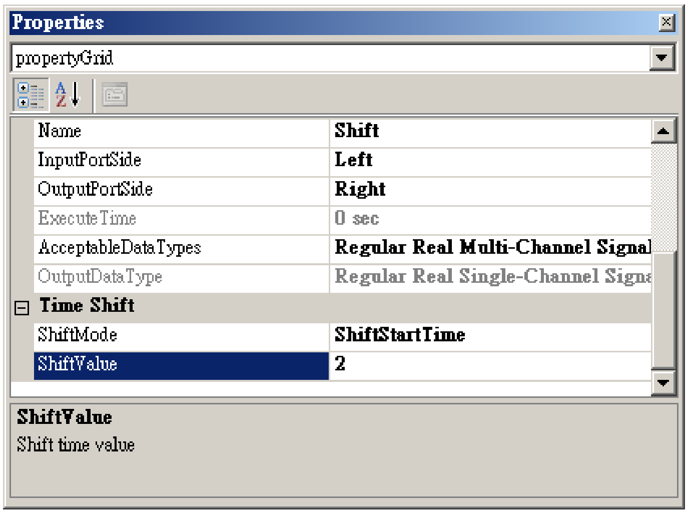
1. Create *Source→Sine Wave* and set the *Properties/TimeStart* to 3. You can see that the first point of the sine wave will begin at the 3 sec mark.

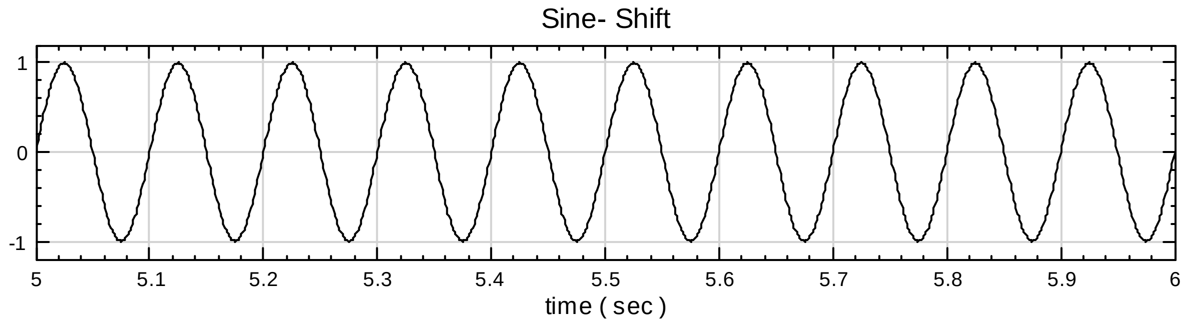




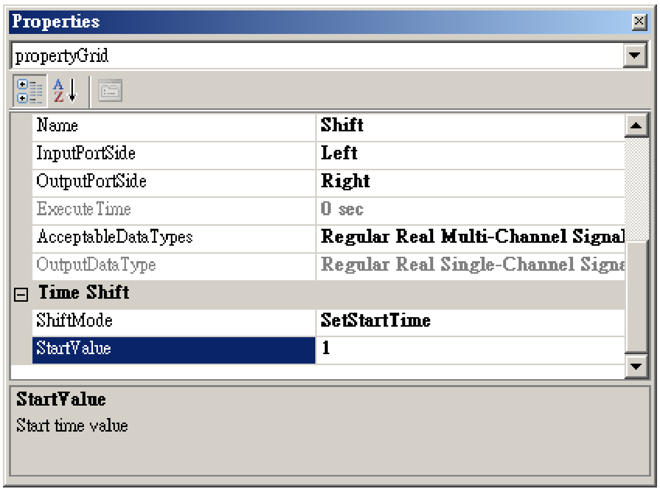
2. Connect the Sine wave SFO to *Compute→Channel→TimeShift* and set the *Properties/ShiftMode* and select *ShiftStartTime* and set *ShiftValue* to 2. You will see that the start time on the graph have shifted to the 5th second.

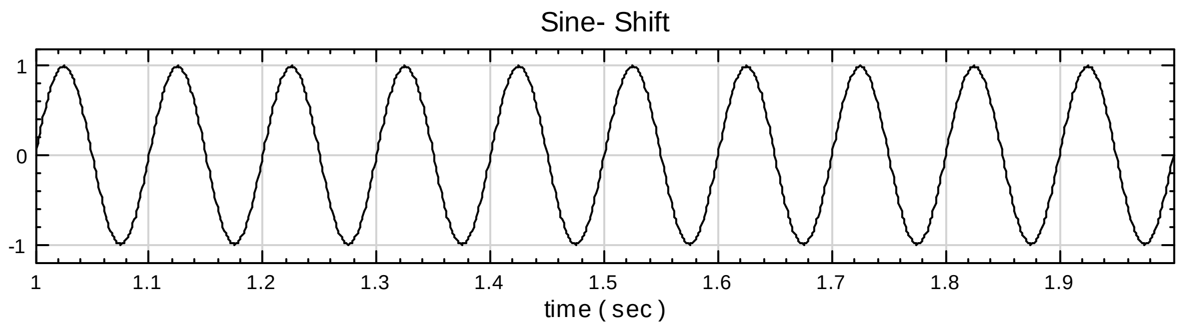






3. If you select *SetStartTime* and set the *StartValue* to 1, then the first point of the graph will start at 1 second mark.





Important

Timeshift allows the user to shift the graph along the x-axis and the RemoveDC allows the user to shift the graph along the y-axis.

Related Functions

Time Shift, RemoveDC.

3.1.9 Data Merge

Connect two one-channel signals together to form a one-chanel signal.

Connect two multi-channel signals together to form a multi-channel signal.

Properties

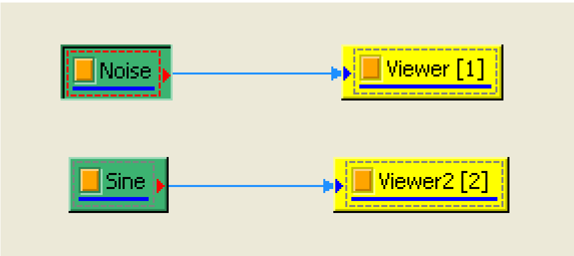
This module accepts input of Signal which could be real number, single channel or multi-channel, regular and audio.

Property Name	Property Definition	Default Value
StartPosition	The new position of the signal to be connected. The new position is relative to the reference signal. The default value is the end position of the reference signal.	end
ReferenceInput	Set the reference signal. StartPosition is relative to the reference signal.	0

Example

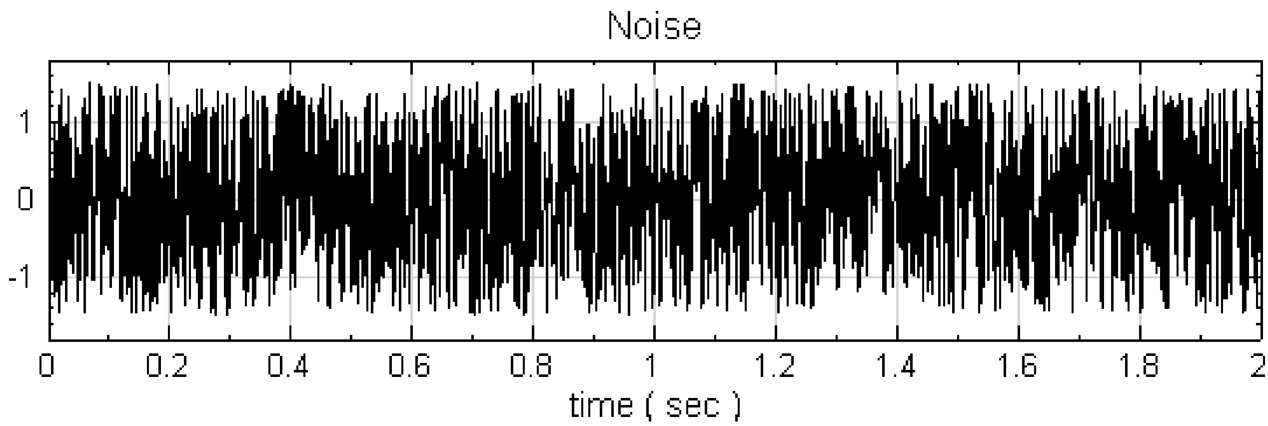
The original siganals are *Source→Noise* and *Source→Sine Wave,* then Data Merge is applied to produce a longer signal for further calculation. The steps are:

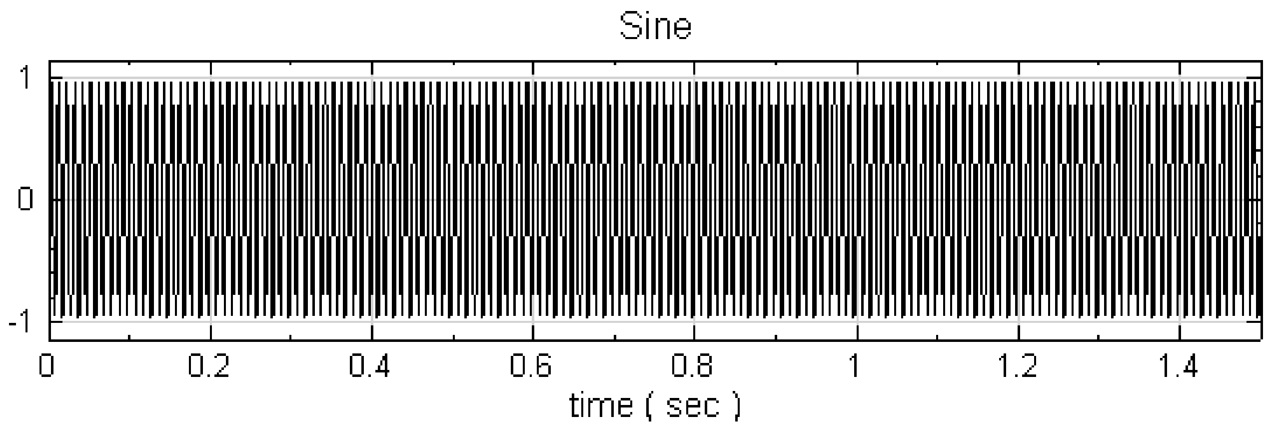
1. Choose *Source→Noise* in Network panel, set *TimeLength* to 2s and *Amplitude* to 1.5. Then choose *Source→Sine Wave,* set *TimeLength* to 1.5s and *SignalFreq* to 100. Both SFOs are connected to *Viewer→Channel Viewer* for display as shown below.



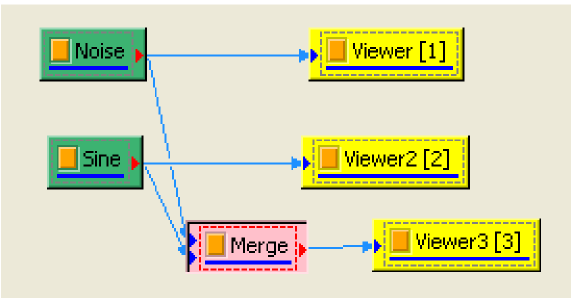
Properties	
⊞ Module	
⊟ Source	
TimeUnit	sec
TimeLength	1.5
SamplingFreq	1000
DataLength	1501
SignalFreq	100
Amplitude	1
AmplitudeOffset	0
Phase	0
TimeStart	0

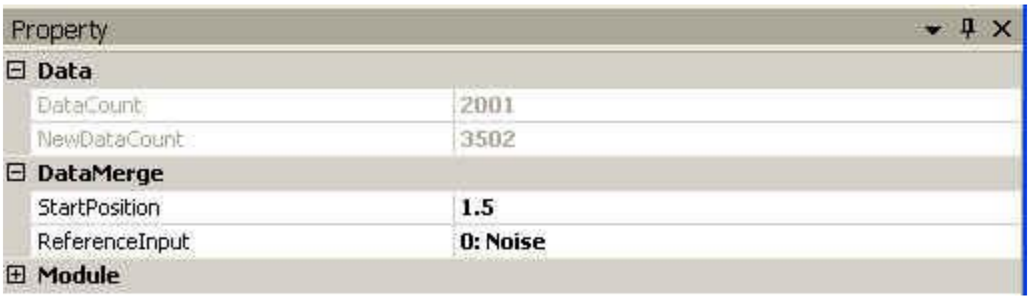
Properties	
⊞ Module	
⊟ Noise	
NoiseType	White
⊟ Source	
TimeUnit	sec
TimeLength	2
SamplingFreq	1000
DataLength	2001
Amplitude	1.5
AmplitudeOffset	0
TimeStart	0

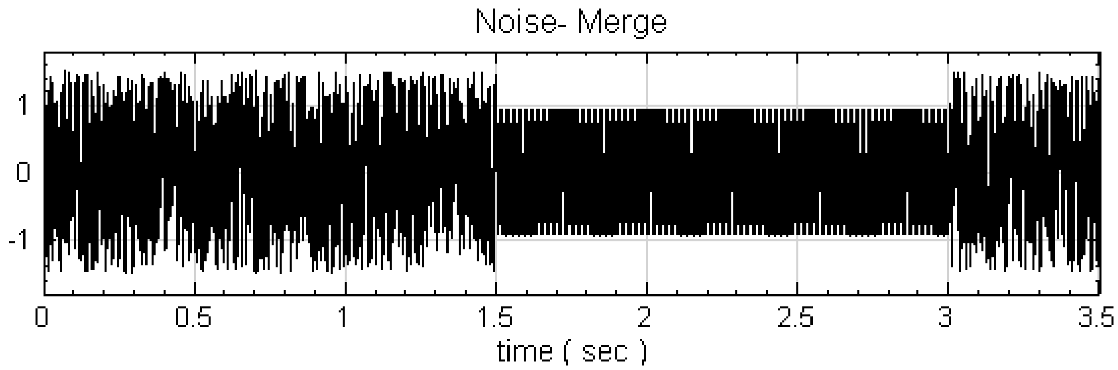




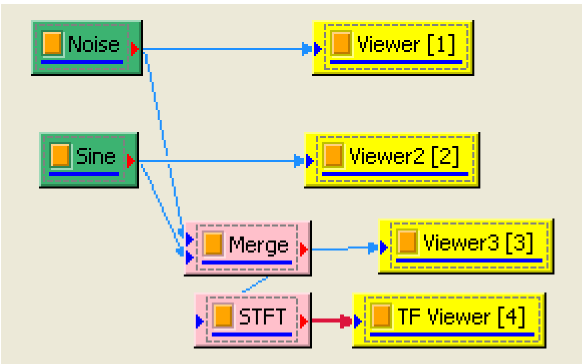
2. Connect both *Noise* and *Sine* SFOs to *Compute→Channel→Data Merge*，set *Data Merge* parameter *ReferenceInput* to 0: Noise，and set StartPosition to 1.5，then connect *Data Merge* to *Viewer→Channel Viewer* for displaying result。

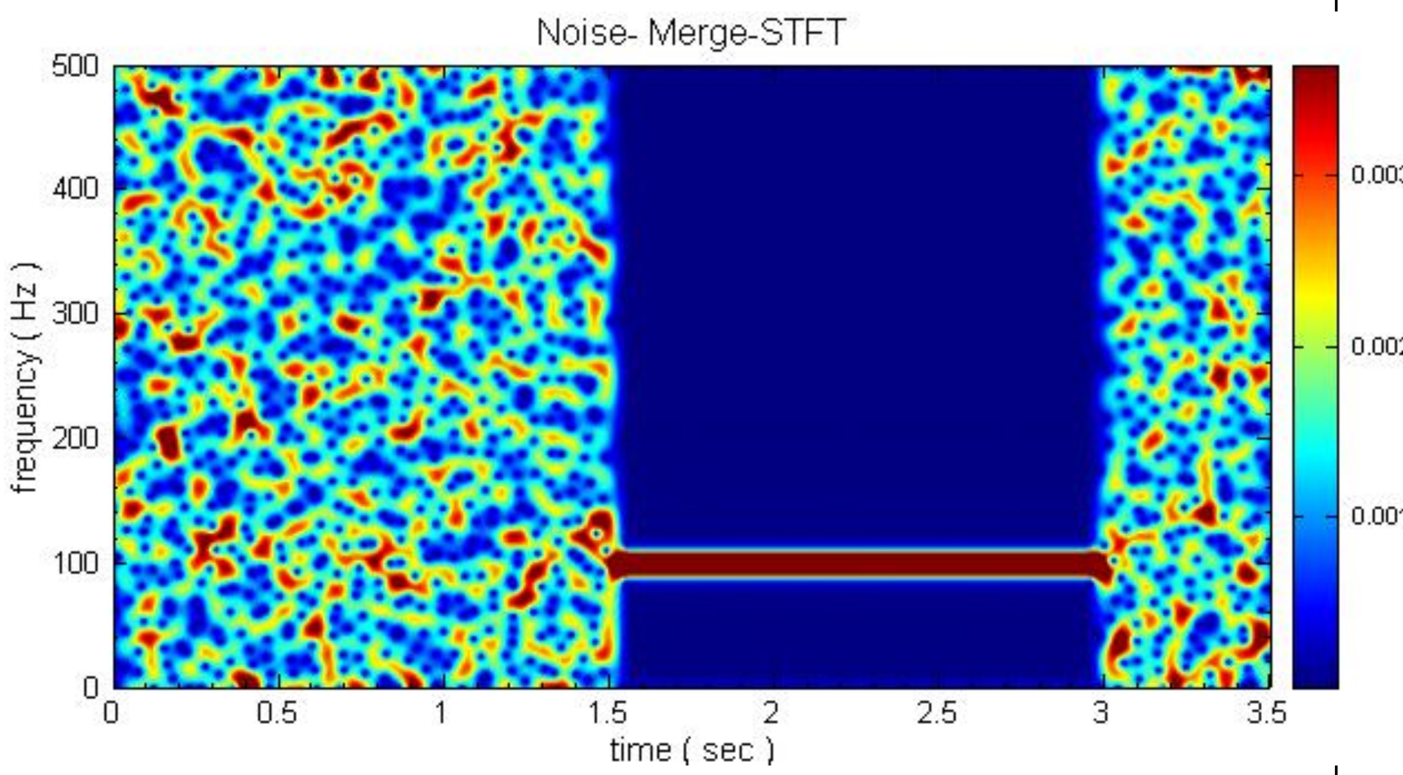






3. Connect *Data Merge* SFO to *Compute→TFA→ShortTerm Fourier Transform* with the default setting， then display the result using *Viewer→Time-Frequency Viewer*. It is shown that the 100Hz Sine Wave is between 1.5s and 3s.





Related Functions

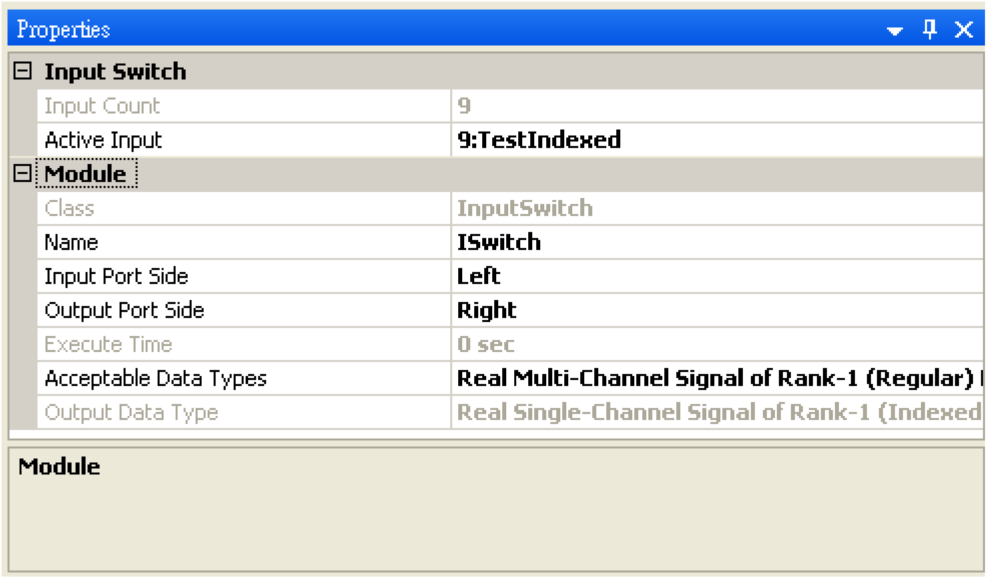
Noise, Sine, ShortTerm Fourier Transform, Channel Viewer, Time-Frequency Viewer.

3.1.10 Input Switch

Select one channel from a multi channel input signal.

Properties

This module accepts input of Signal which could be real number or complex, single channel or multi-channel, regular, indexed, audio, numeric, and spectra. The definition and default vaule of the parameters are shown below.

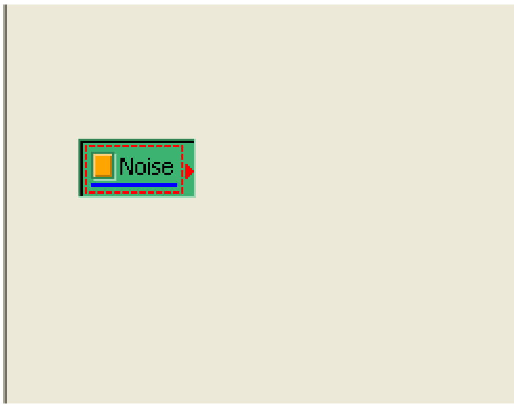


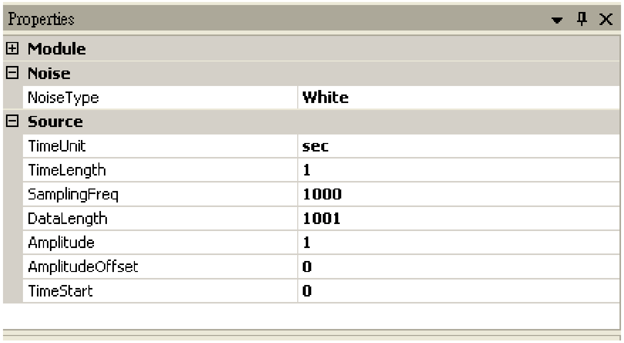
Property Name	Property Definition	Default Value
Input Count	Total number of channels in this SFO	0
Active Input	The selected channel number	1

Example

Use *Source→Noise* as the original signal, carry out different calculations on it and output the results in different format. Then use *Input Switch* to select one of the channels. Steps are shown below.

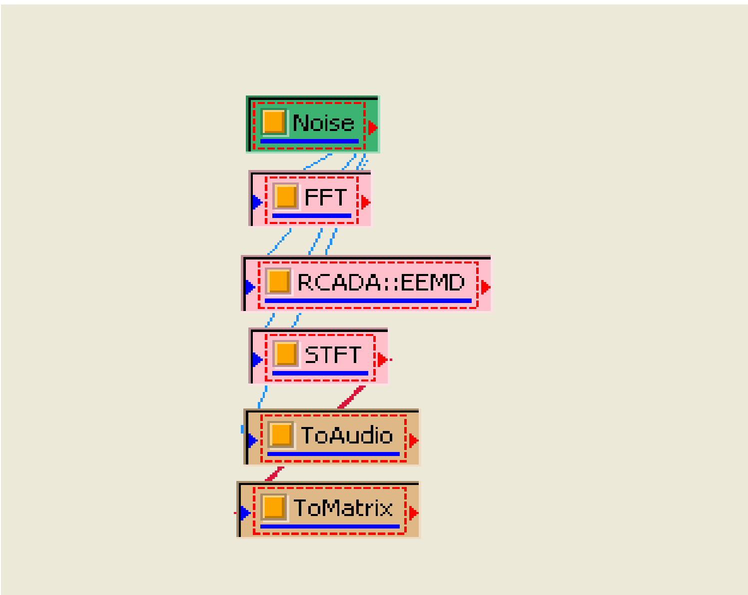
1. Create Source→Noise in Network panel, set Properties/TimeLength to 3, set Properties/SamplingFreq to 1000, and set Properties/Amplitude to 1.



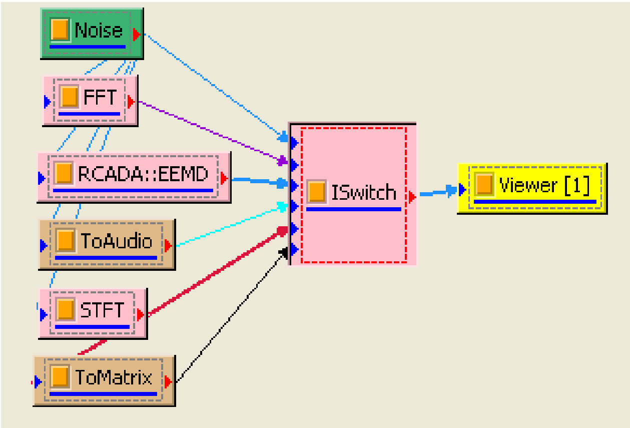

Properties

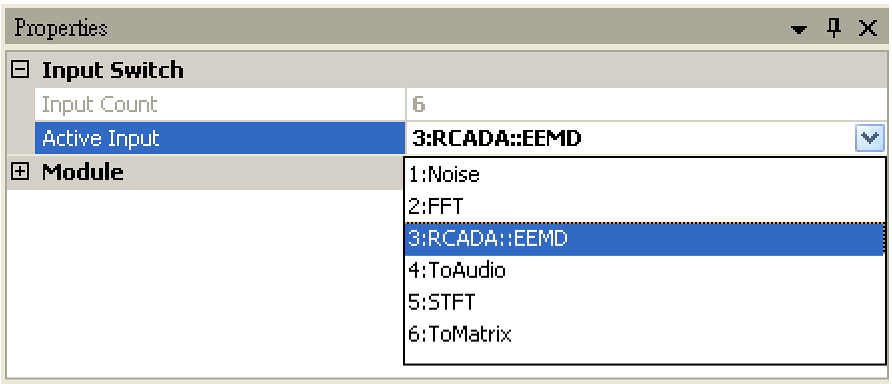
Module	
Noise	
NoiseType	White
Source	
TimeUnit	sec
TimeLength	1
SamplingFreq	1000
DataLength	1001
Amplitude	1
AmplitudeOffset	0
TimeStart	0

2. With default settings, connect the *Noise* SFO to *Compute→Transform→Fourier Transform*, *Compute→Transform→RCADA EEMD*, *Conversion→Convert to Audio*, *Compute→TFA→Short Term Fourier Transform*, and *Conversion→Convert to Matrix* respectively.

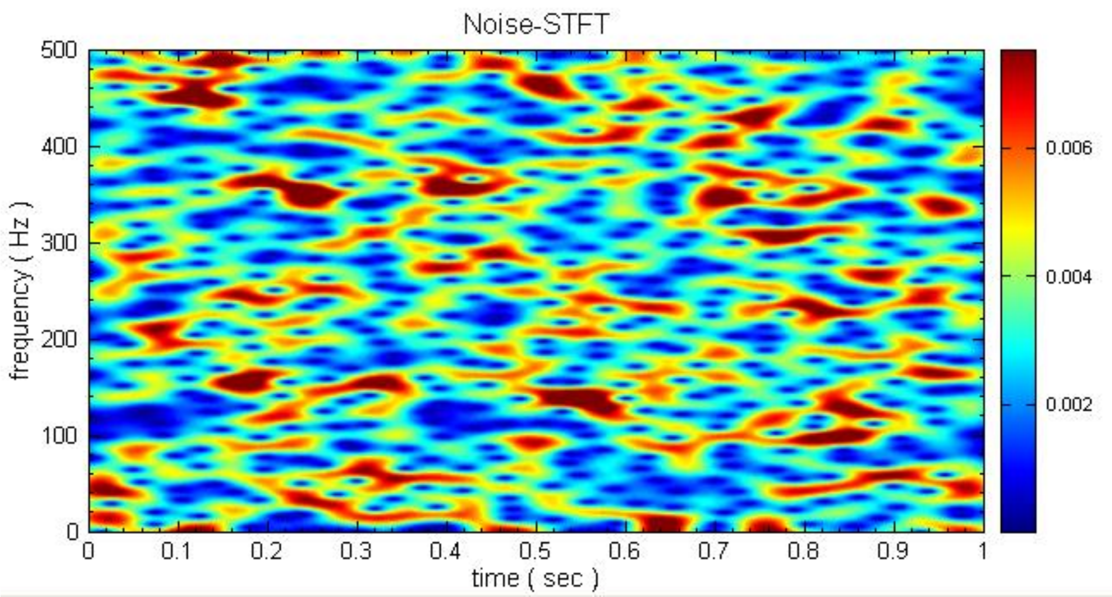


3. All outputs of the calculation are connected to *Compute→Channel→Input Switch,* and show the result in *Channer Viewer*. Change *Active Input* setting in *Input Switch* to view different results.

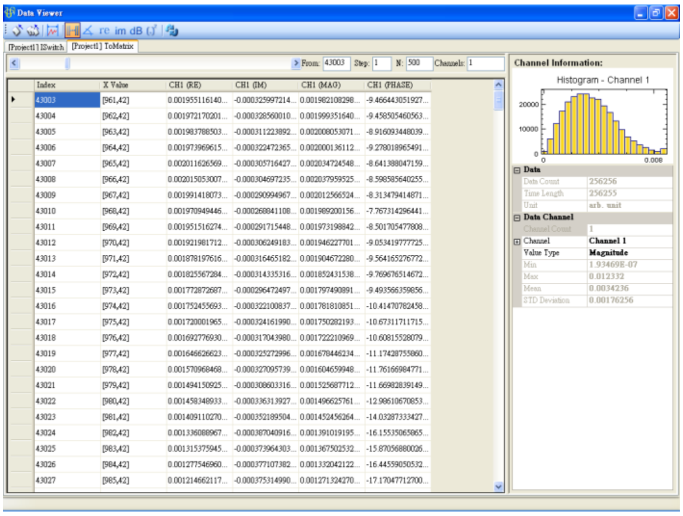




4. And connect *Input Switch* to *TFA Viewer* and change *Active Input* setting to observe the result of STFT.



5. Change *Active Input* setting to ToMatix and use *DataViewer* to view the matrix content of *Input Switch* SFO.



Related Functions

Noise, Fourier Transform, RCADA EEMD, Convert to Audio, ShortTerm Fourier Transform, and Convert to Matrix.

3.2 Filter

This module provides several regular filters which are used to remove some components from input signal, based on different signal characteristics.

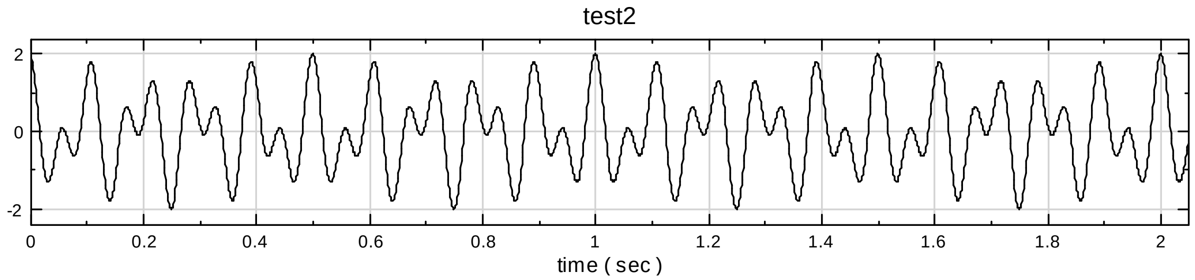
1. **FIR Filter:** Fundamental Finite Impulse Response Filter.
2. **Median Filter:** Significantly reduce impulsive noises.
3. **Moving Average Filter:** Used to remove the random noise.
4. **Iterative Gaussian Filter:** Efficiently remove aperiodic components from an input signal.
5. **Trend Estimator:** Simplified version of Iterative Gaussian Filter. Used to extract aperiodic components from an input signal.

3.2.1 FIR Filter

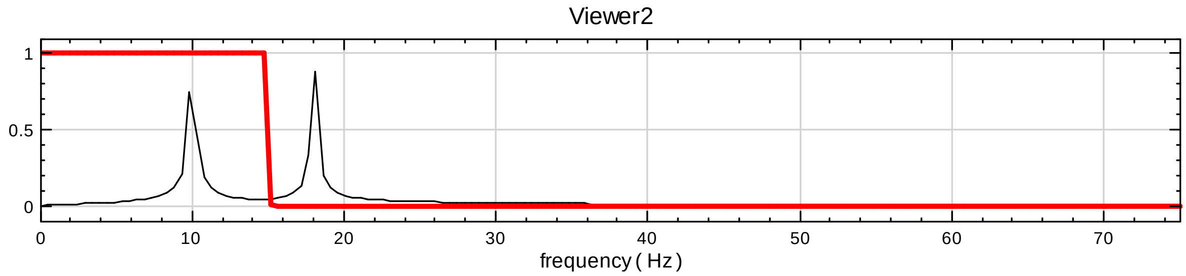
Finite Impulse Response Filter is the fundamental filter prototype in digital signal processing, which can remove high-frequency, low-frequency or a given band frequency components. The term of *finite* means that the filter impulse response is finite.

Introduction

Assume an input signal is given as shown below.

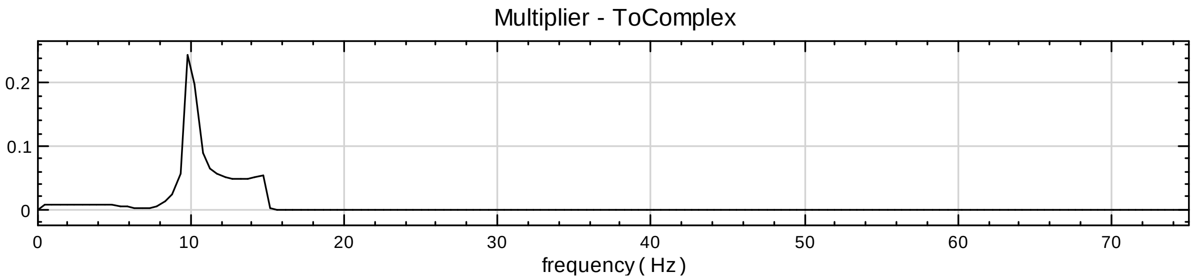


The Fourier Transform is shown below. It is desired to remove the high frequency components and preserve the low frequency components.



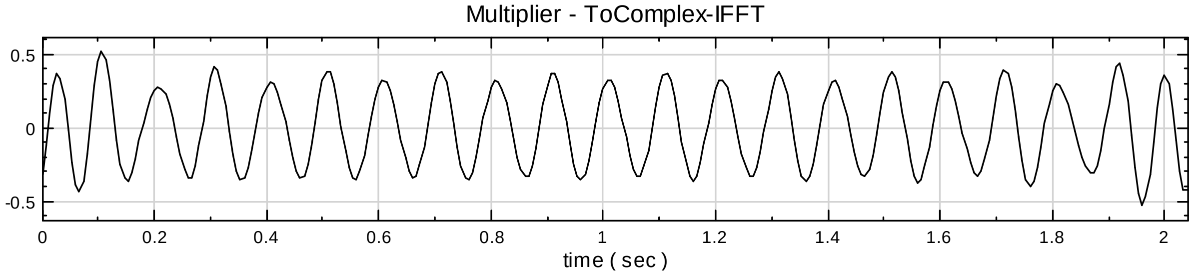
(The thin black curve represents the Fourier Transform of the original signal and the bold red curve represents the desired filter)

Therefore, define a function representing the filter above in Fourier Space and multiply it with the Fourier Transform of the original signal.

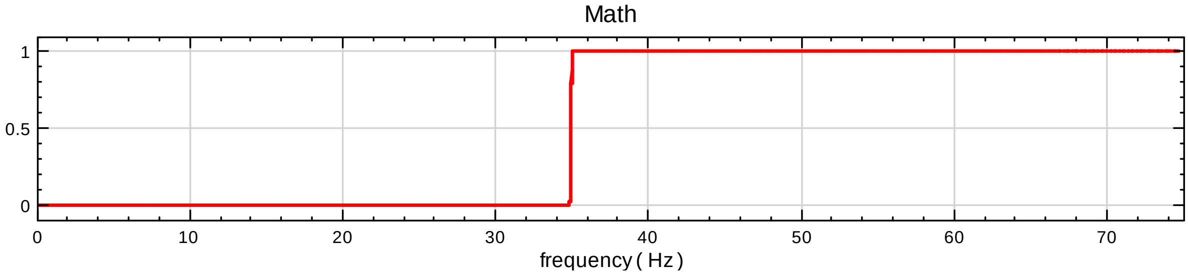


Next, conduct Inverse Fourier Transform to remove the high frequency

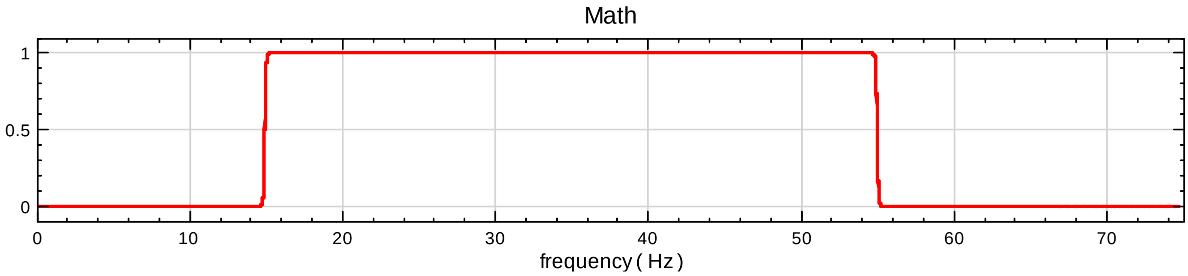
component. The result is shown below.



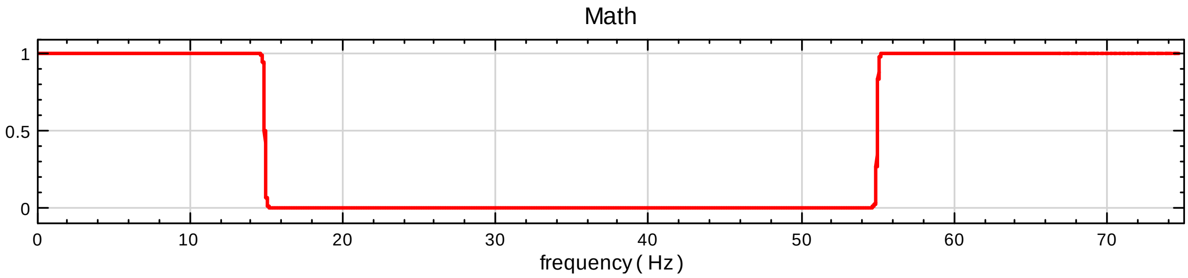
Besides the low-pass filter above, the high-pass filter is shown below.



BandPass: The BandPass filter is shown below.



BandStop: The BandStop filter is shown below.



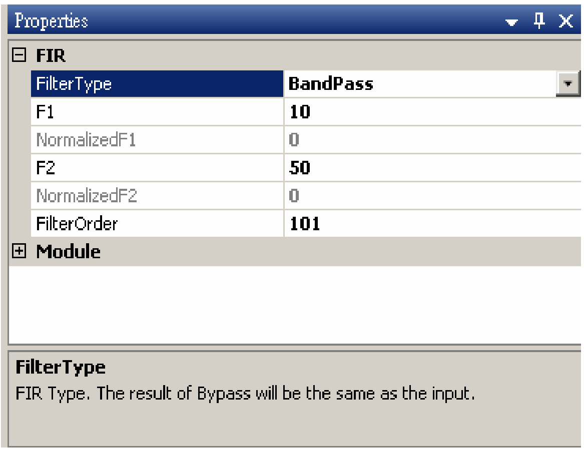
Bypass: All frequency components can pass through the filter.

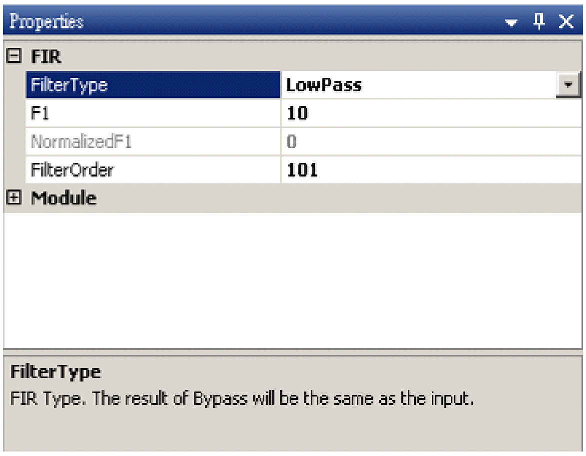
Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or

multi-channel, Regular).

The main property of FIR Filter is FilterType, which has 5 options: LowPass, HighPass, BandPass, BandStop and ByPass. LowPass is used to remove frequency components which are higher than F1, while HighPass is used to remove components which are lower than frequency F1. BandPass is used to retain components which are between frequency F1 and F2 while BandStop is used to remove them. ByPass allows all components to pass through, i.e., the output signal is the input signal. Definition of properties and default values are shown below.





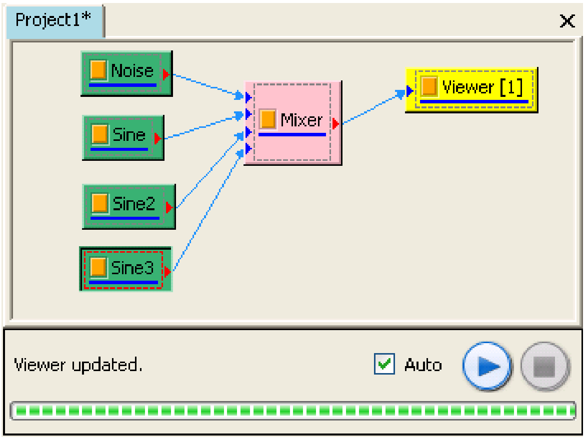
Property Name	**Property Definition**	**Default Value**
FilterType	5 types are provided which are LowPass, HighPass, BandPass, BandStop, and ByPass.	LowPass
F1	For LowPass and HighPass, F1 represents the cutoff frequency. For BandPass and BandStop, F1 represents the frequency starting point. Unit is Hz.	10
NormalizedF1	Demonstrate the normalized F1 based on the Sampling frequency of the input signal.	Varies based on the input signal
F2	The frequency ending point, F2, for BandPass and BandStop filters. Unit is Hz.	50

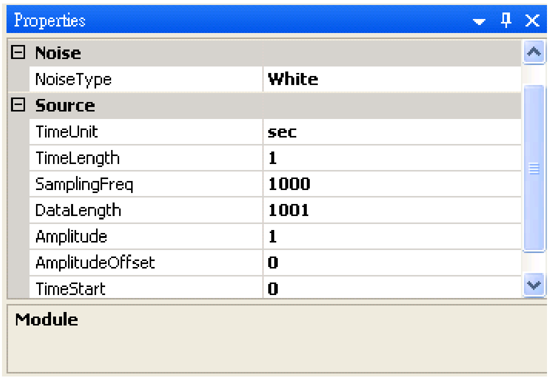
FilterOrder	The number of points in the discrete impulse response function of the filter. N means N-order Filter.	101

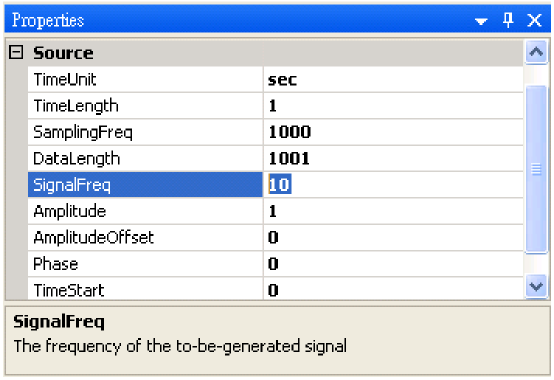
Example

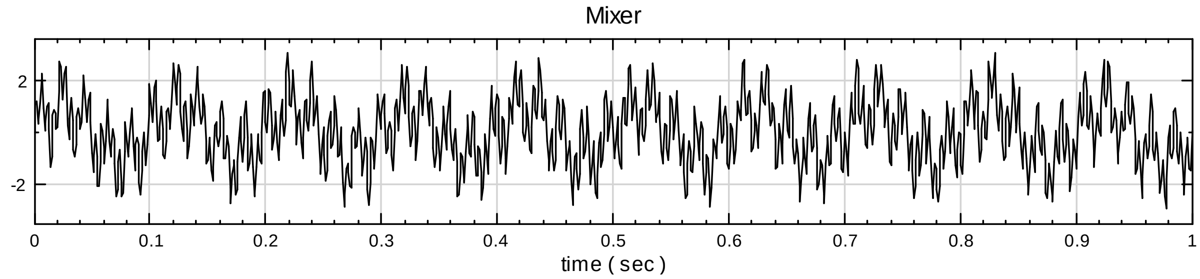
This example shows the process of using FIR Filter to remove different frequency components based on an input signal which contains 10, 51, 193 Hz sine waves plus white noise.

1. In the Project window, select *Source→Noise* to create a white noise signal and set the *Properties/Amplitude* as 0.3. Then use the *Source→Sine Wave* to generate 3 sine waves and change their *Properties/SignalFreq* to 10, 51, 193 Hz. After that, use the *Compute→Mathematics→Mixer* to mix the above signals and plot them using the *Viewer→Channel Viewer* (by dragging the *Output* of every signal to the Input of *Mixer*).

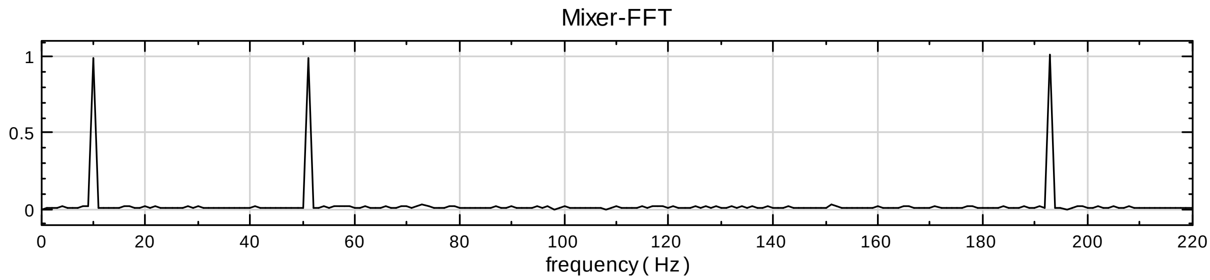




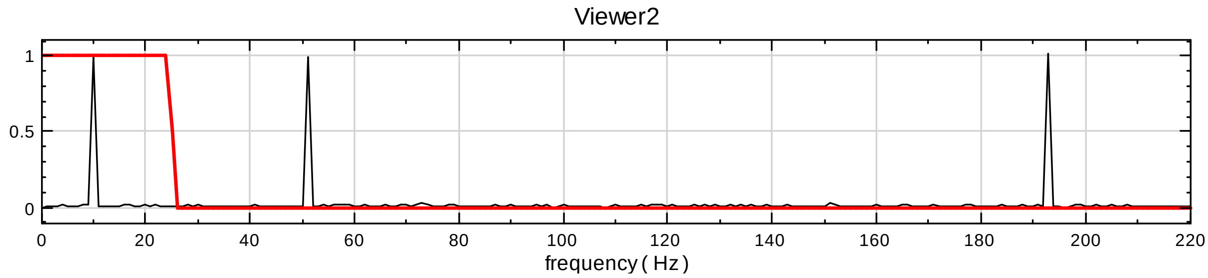


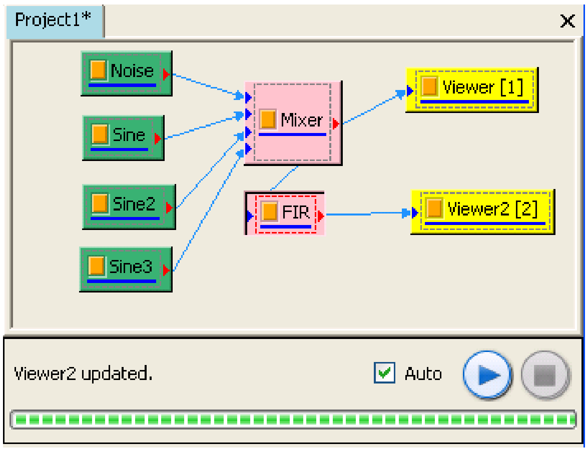


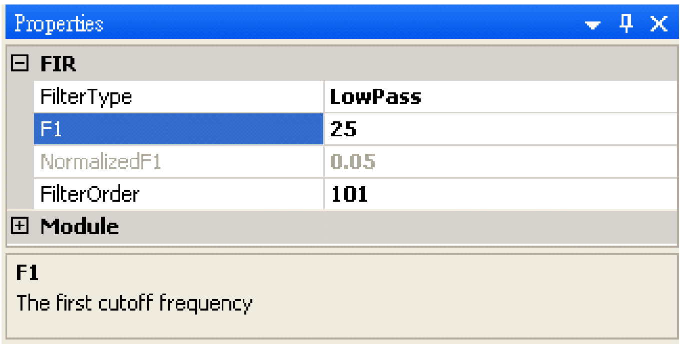
(To facilitate the following FIR Filter design, *Mixer* could be connected to FFT for frequency spectrum observation).

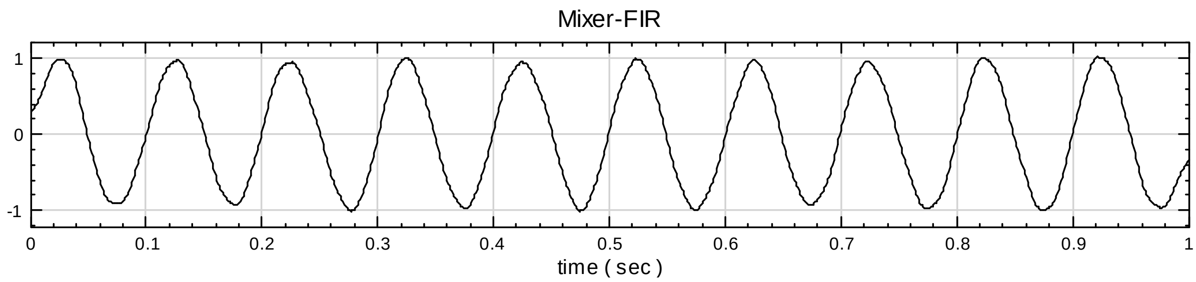


2. On the *Mixer* icon, select *Compute→Filter→FIR* Filter and change the *Properties/F1* to 25Hz. The default *FilterType* is *LowPass*. Then, use *Channel Viewer* to show the processing result. It can be seen that the frequency components higher than 25Hz are all removed and the output signal is similar to sine of 10Hz. However, because the Filterorder is only 101, the wave is partially affected.

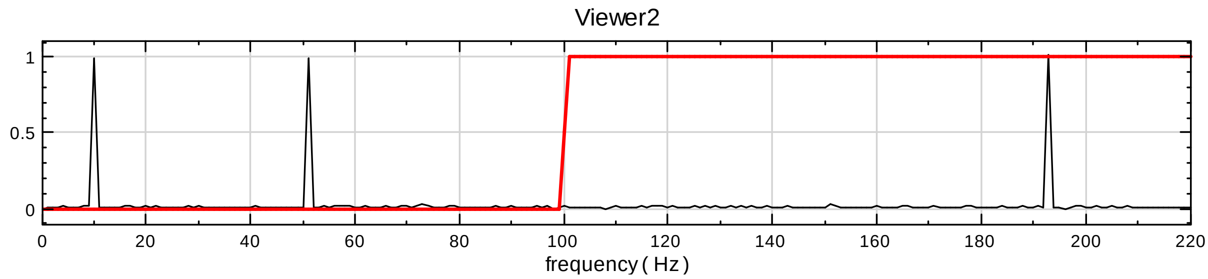


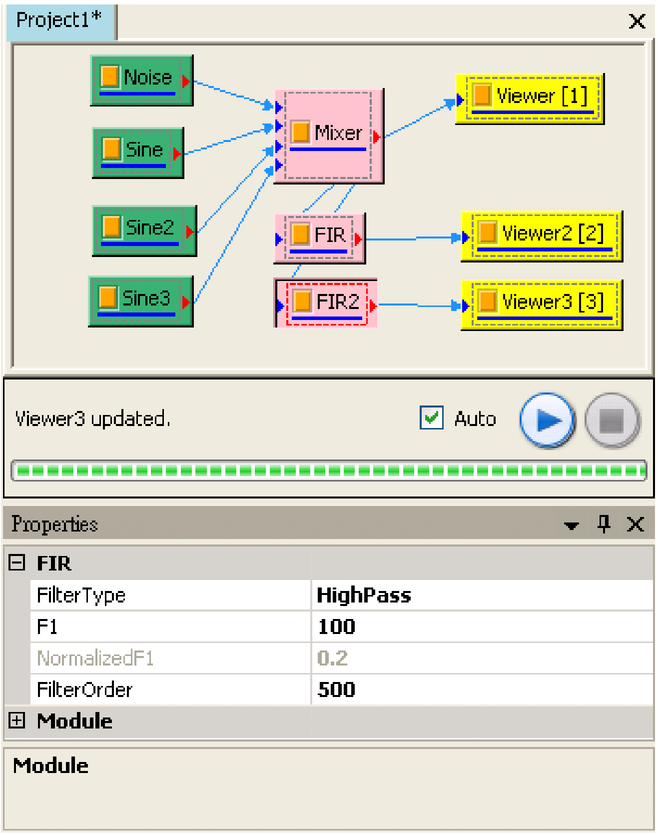


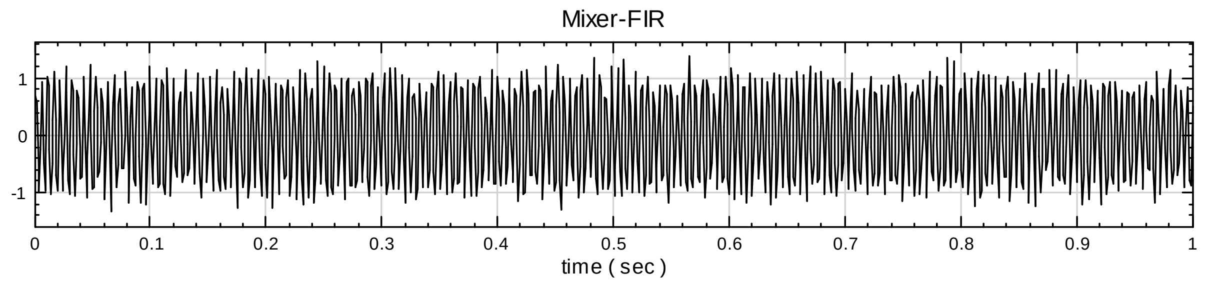




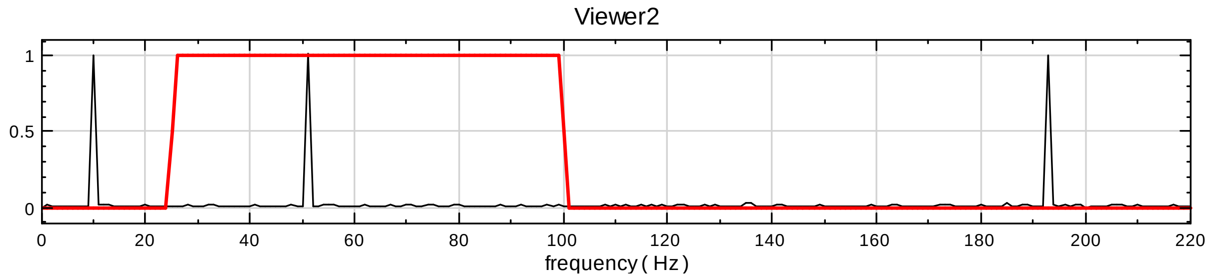
3. Change properties of *FilterType* to *HighPass*, F1 to 100Hz, and *FilterOrder* to 500. As shown below, the filter removes the frequency component lower than 100Hz and it leaves a sine wave of 193Hz with the White Noise.

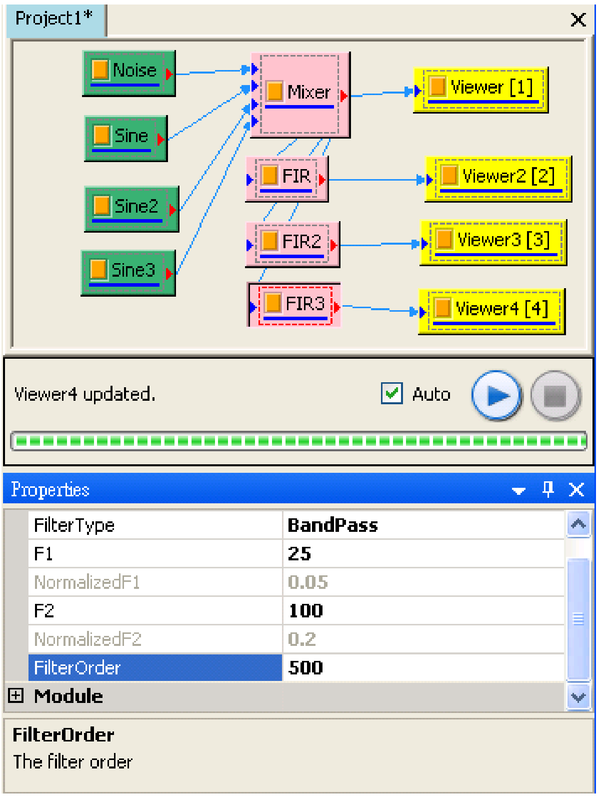


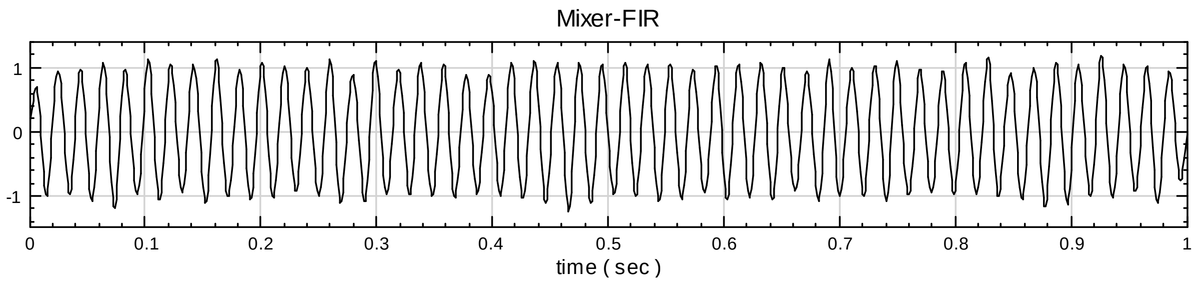




4. Repeat step2 and add one FIR filter. Change the FilterType to BandPass, F1 to 25Hz, F2 to 100Hz, and FilterOrder to 500. The frequency components between 25~100Hz would pass through while other components would be cut off. Therefore, the output is a sine of 51Hz.







Related Functions

Noise, Sine, Mixer

References:

http://en.wikipedia.org/wiki/Finite_impulse_response

http://cnx.org/content/m11918/latest

3.2.2 Median Filter

Median Filter is a one-dimensional non-linear filter, used to calculate the median in the range of filtering (Filter Order). Because it can reduce speckle noise significantly while retain good edge detection, it is usually used in digital image processing.

Introduction

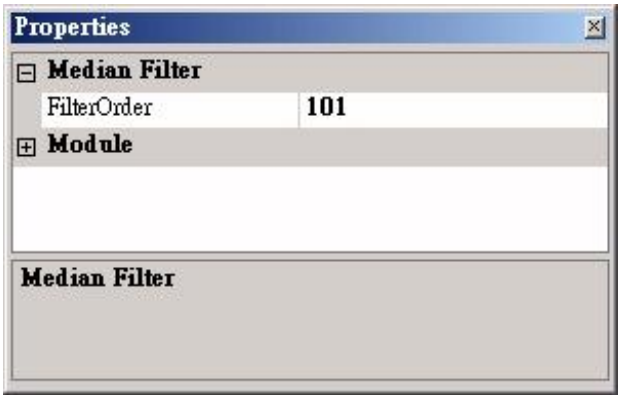
Let $X = \{x_0, x_1 \ldots x_{N-1}\}$ be a N-length input signal, $Y = \{y_0, y_1 \ldots y_{N-1}\}$ be the output signal, and *M* be the signal length for calculation. The *Median Filter* is defined as

$$y_i = median(x_k),\ i - \frac{M-1}{2} \le k \le i + \frac{M+1}{2}$$

Centered on the i^{th} data, take $\frac{M-1}{2}$ points on both sides to construct a set of array. Then, find the median in the array to replace the i^{th} data. In the case when the number of data is insufficient, e.g., $M > N$ or $N - i < \frac{M-1}{2}$, repeat the edge data to fill the whole array. *M* is supposed to be an odd number. In the case of even number, it would be made to be odd by adding 1 automatically.

Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular). The formats of input signal and output signal are identical.

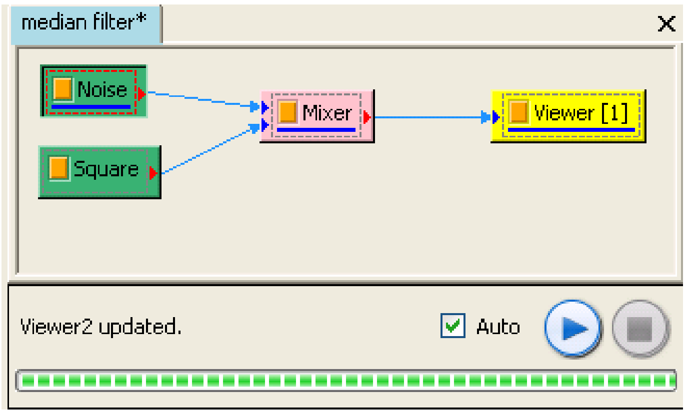


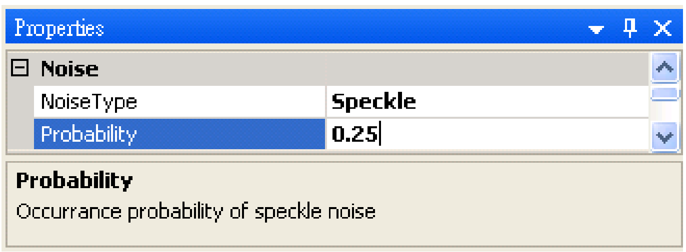
Property Name	Property Definition	Default Value
FilterOrder	The data length of the median filter, i.e., *M,* is supposed to be an odd number. In the case of even number, it would be changed to an odd number by adding 1 automatically.	101

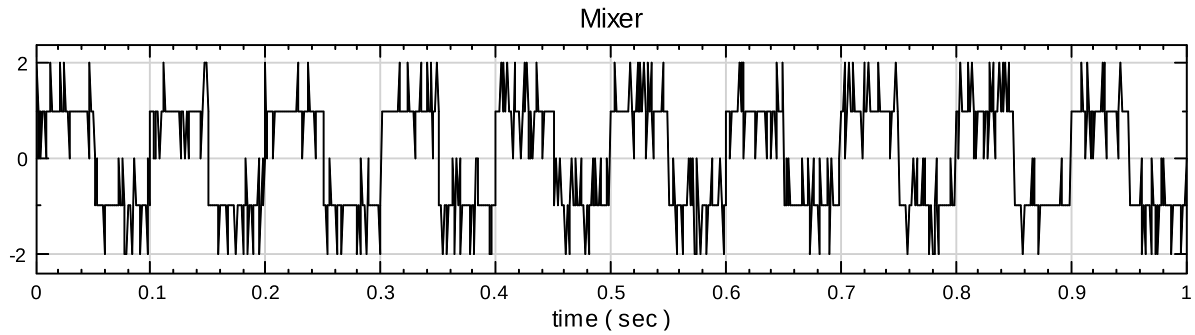
Example

The example shows the procedure of using a median filter to process a signal of a square wave plus speckle noise.

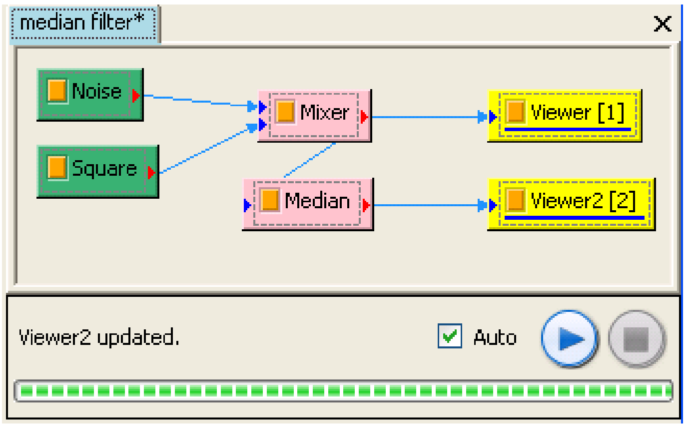
1. Right click in the Network Window, select *Source→Noise* to generate a noise signal. Set the *Properties/NoiseType* as *Speckle*, *Probability* as 0.25. In addition, select *Source→Square Wave* to generate a square wave. Use *Compute→Mathematics→Mixer* to mix these two signals and then use *Viewer→Channel Viewer* to show the result in the window.

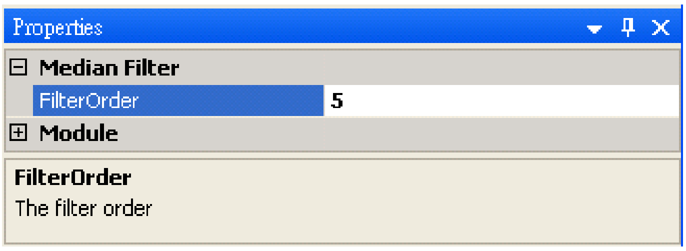


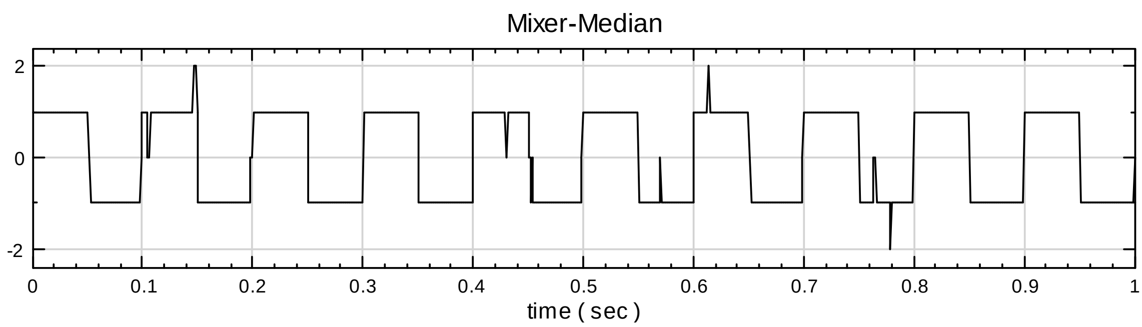




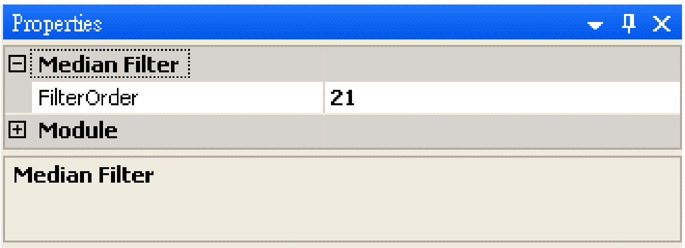
2. Click on the *Mixer* icon to select *Computer→Filter→Median Filter*, change the *Properties/FilterOrder* to 5, and then use *Channel Viewer* to show the result.

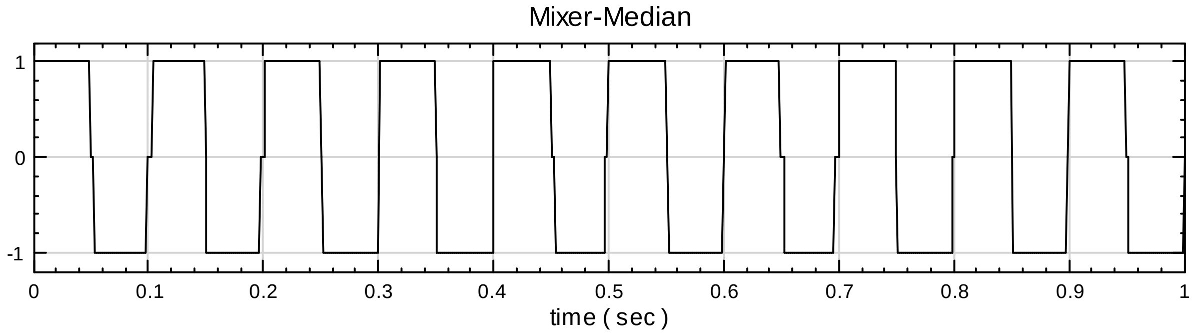




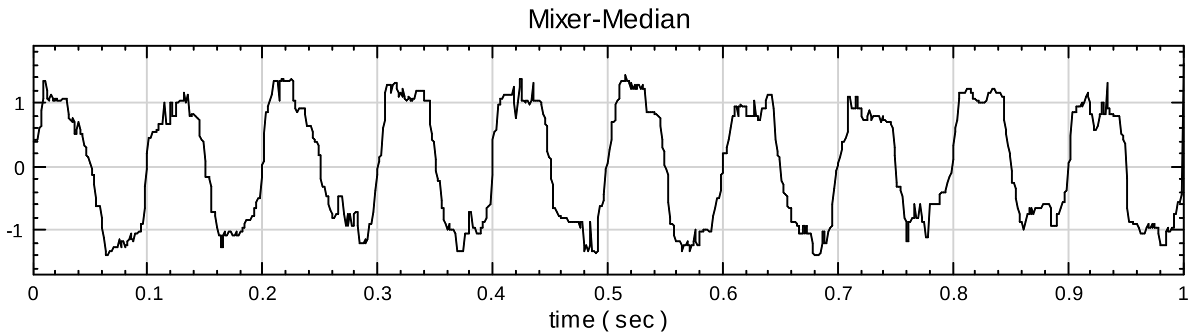


3. In step 2, good result is achieved. As shown below, *FilterOrder* could be increased 4 times if the Median Filter is adjusted to 21, i.e., FilterOrder = 21. Not only is the speckle noise removed completely, but also the wave preserves good edge sharpness.





4. Last step, to test characteristics of Median filter, come back to *Noise* to change *Speckle noise* to *White noise* in *NoiseType*. As shown below, it can be seen that the Median Filter cannot eliminate the effect caused by White noise completely. However, the wave edges are mostly retained. This is the main characteristic of Median filter.



Related Functions

Noise, Square, Mixer, Moving Average Filter.

Reference

http://en.wikipedia.org/wiki/Median_filter

3.2.3 Moving Average Filter

By calculating the average of signals in the range of filtering (average length), *Moving Average Filter* decreases the noise in discrete time signals and increase the recognizability of peak. The advantages of moving average filter are: simple theory and fast calculation. However, compared with other types of filters, it has a low filtering ability to separate one band of frequencies from another. In spectrum analysis, its performance is poor.

Introduction

Let $X = \{x_0, x_1 ... x_{N-1}\}$ be an N-length input signal, $Y = \{y_0, y_1 ... y_{N-1}\}$ be the output. If the average length of the signal is M elements, for every signal in *X*, the output is

$$y_i = \frac{1}{M} \sum_{j=\frac{-(M-1)}{2}}^{\frac{M-1}{2}} x_{i+j}$$

The formula above means the convolution of the input signal and a square filter which has area of 1 and length M in time axis.

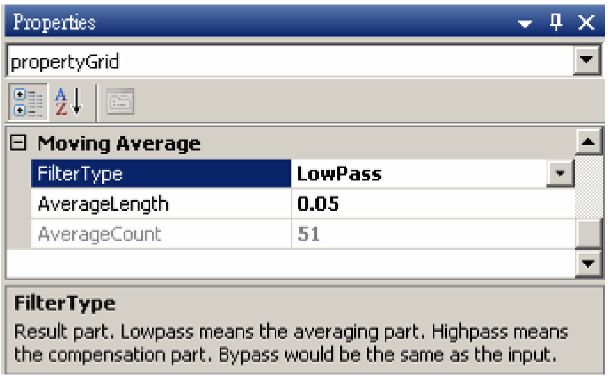
Notice that this filter is similar to the Rolling Statistics on the average calculation. The difference is how to handle with the edge. On the edge, in the case when average length M_b is less than *M*, this filter still calculates average using M_b. Therefore, the output data length is identical to the input data length. On the other hand, the Rolling Statistics only calculates average in the range of given data and therefore, the length of output data length would be less than the length of input data.

Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular). The input signal format and the output signal format are identical.

Moving Average Filter has two properties, which are FilterType and AverageLength. AverageLength represents the number of data points, *M*, for average calculation. The unit is time. FilterType sets the type of filters which include LowPass, HighPass and ByPass. LowPass is the calculation result using the theory introduced

above. HighPass is achieved by subtracting the LowPass result from the input signal. Because the original signal is equal to HighPass + LowPass, the output of ByPass is equal to the input signal. The default values of properties are shown in the table below.

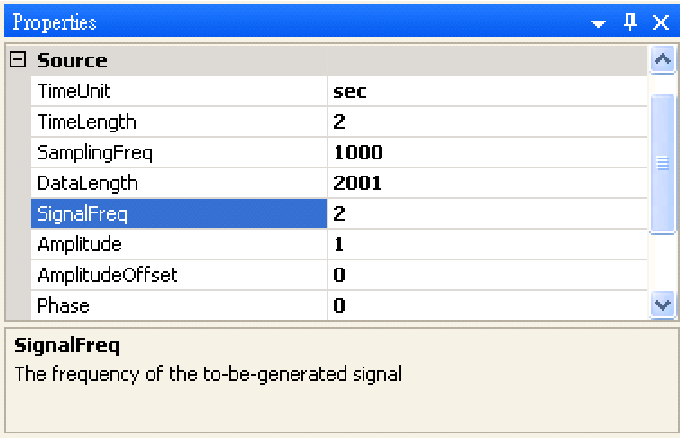


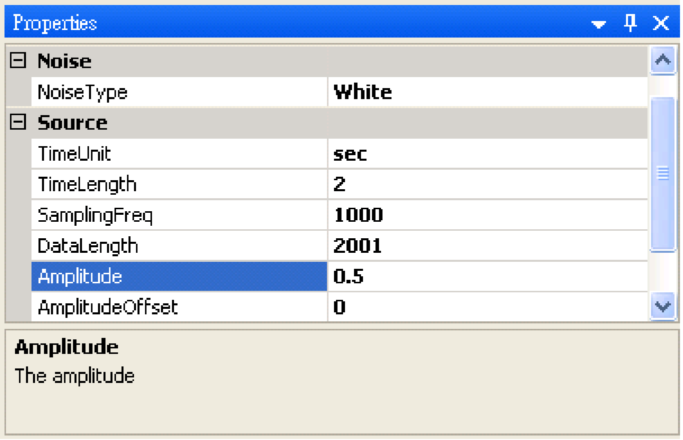
Property Name	Property Definition	Default Value
FilterType	To set the filter to remove high-frequency or low-frequency components. The available options are LowPass, HighPass, and ByPass.	LowPass
AverageLength	The signal length for calculation of average. The unit is time.	0.05 of total length
AverageCount	To show the number of signals corresponding to AverageLength	Automatically adjust based on AverageLength.

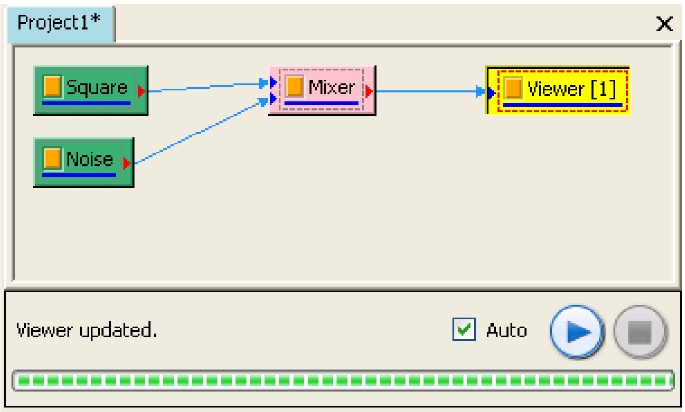
Example

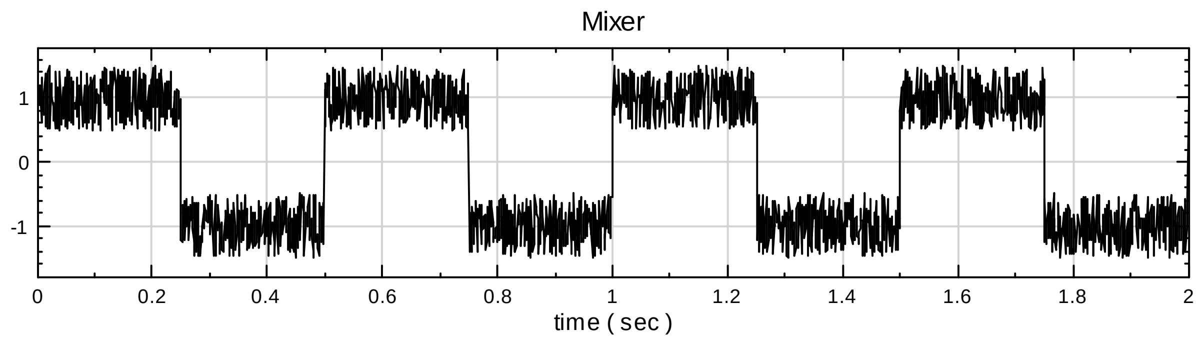
In this example, a square wave (frequency = 2Hz, amplitude=1, duration = 2s) is mixed with a *White Noise* (amplitude=0.5, duration = 2s) and then processed by the *Moving Average Filter.* Different AverageLengths are set to observe corresponding effects.

1. Right click and select *Source→Square Wave* to create a square wave. Change its *Properties/TimeLength* to 2 and *SignalFreq* to 2. Right click again and select *Source→Noise→White Noise* to generate White Noise. Change both *White Noise* and *TimeLength* to 2 and *Amplitude* to 0.5. Finally, right click and use *Compute→Mathematics→Mixer* to mix these two signals and use *Viewer* to plot the result.

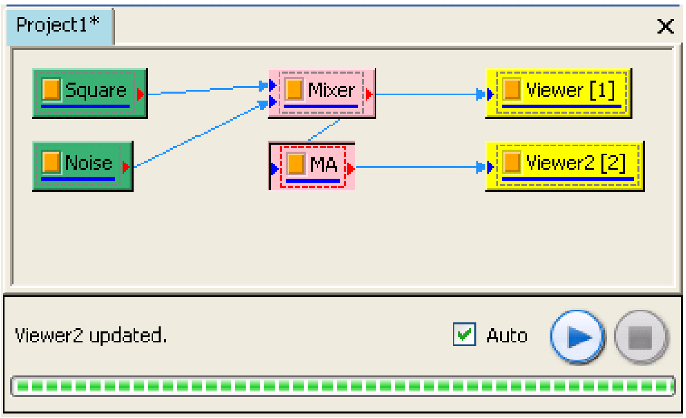

Properties
Source
TimeUnit sec
TimeLength 2
SamplingFreq 1000
DataLength 2001
SignalFreq 2
Amplitude 1
AmplitudeOffset 0
Phase 0
SignalFreq
The frequency of the to-be-generated signal

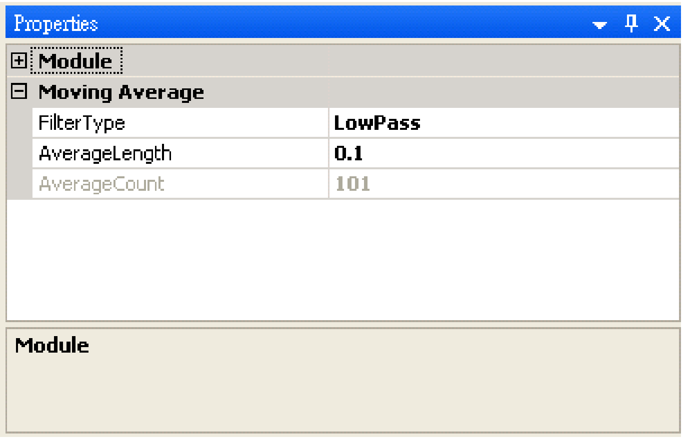

Properties
Noise
NoiseType White
Source
TimeUnit sec
TimeLength 2
SamplingFreq 1000
DataLength 2001
Amplitude 0.5
AmplitudeOffset 0
Amplitude
The amplitude

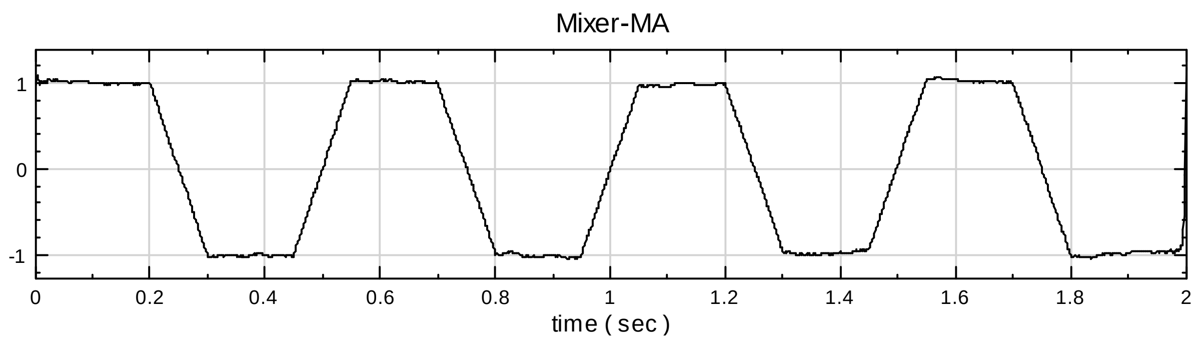

Project1*
Square
Noise
Mixer
Viewer [1]
Viewer updated.
Auto


Mixer
1
0
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
time (sec)

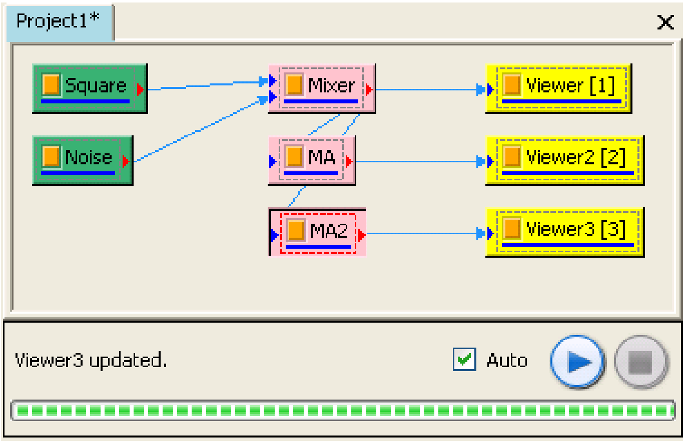
2. To conduct *Moving Average* on the input signal, right click on *Mixer* and select *Computer→Filter→Moving Average.* In the field of *Properties/AverageLength*, it can be seen that the default value is 0.1s. Similarly, it can be seen that the value of *AverageCount* is 101 which means that every output point of MA is the average of 101 points centering at the corresponding point in the input signal. Finally, use *Viewer* to plot the result.

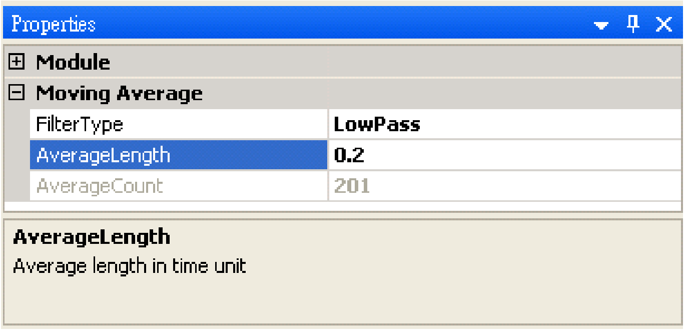


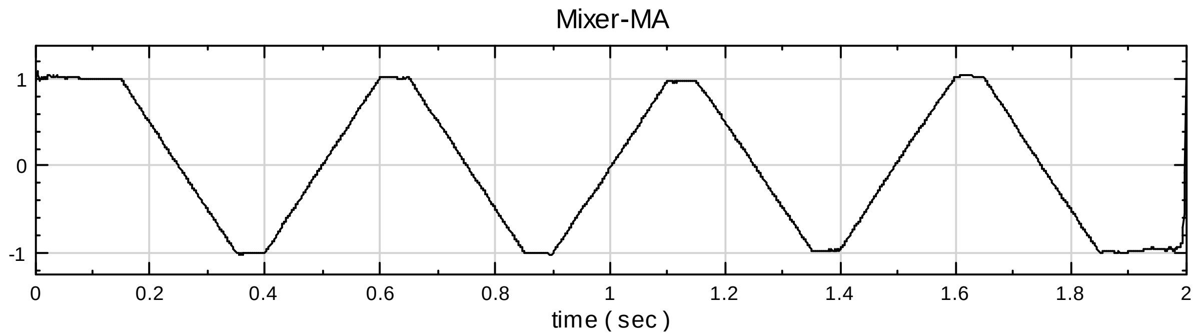




3. Following step 2, change *AverageLength* to 0.2 to perform *Moving Average* again to generate a new figure, named MA2. And use *Viewer* to plot the result.

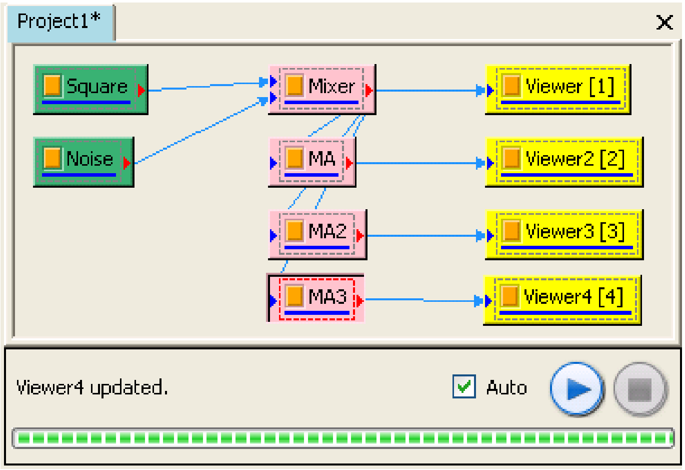


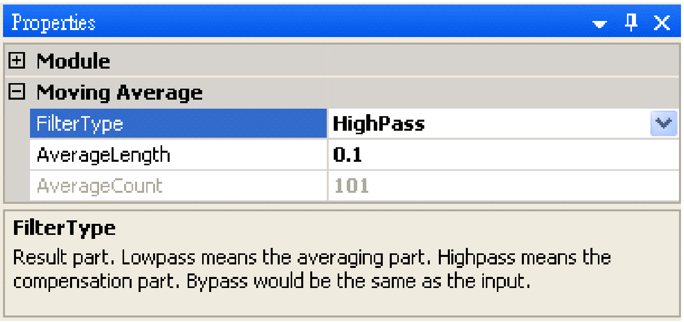


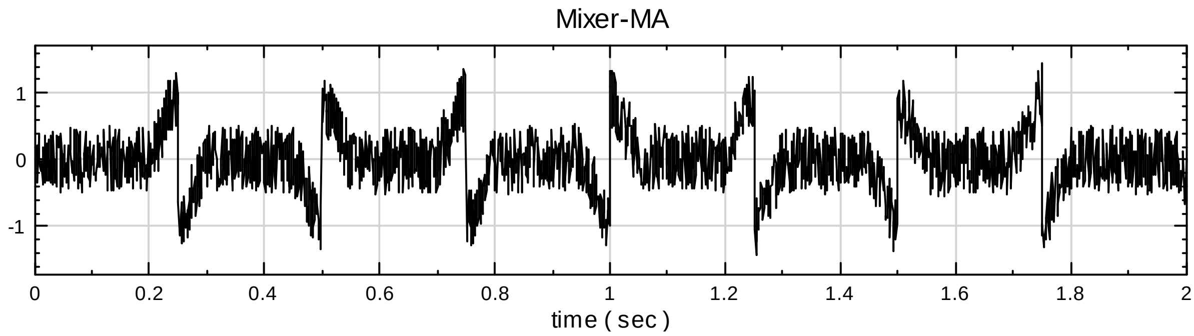


4. Comparing the results obtained in step 1, 2 and 3, it can be seen that increasing *Average Length* can reduce the noise in the input signal significantly. However, the drawback of this filter is that the sharp edge of the original square wave becomes more and more flat as the *Average Length* increases.

5. Next, after changing the *FilterType* to *HighPass,* perform *Moving Average* again. It can be seen that the result is the input signal subtracted by the output of MA2.







Related Instructions

Square, Noise, Mixer.

Reference

http://www.dspguide.com/ch15.htm

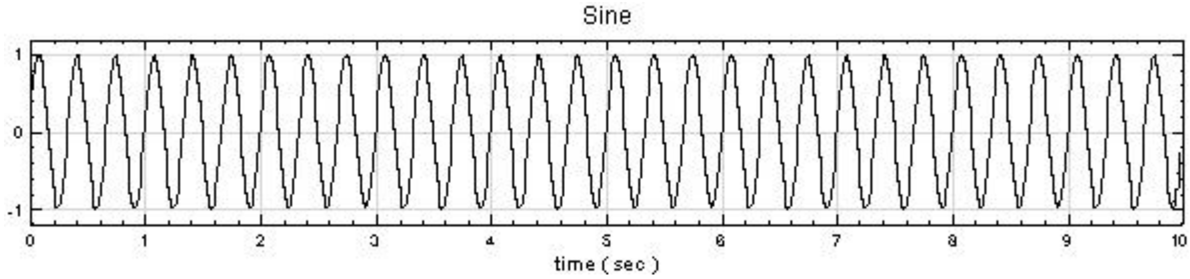
3.2.4 Iterative Gaussian Filter (Professional Only)

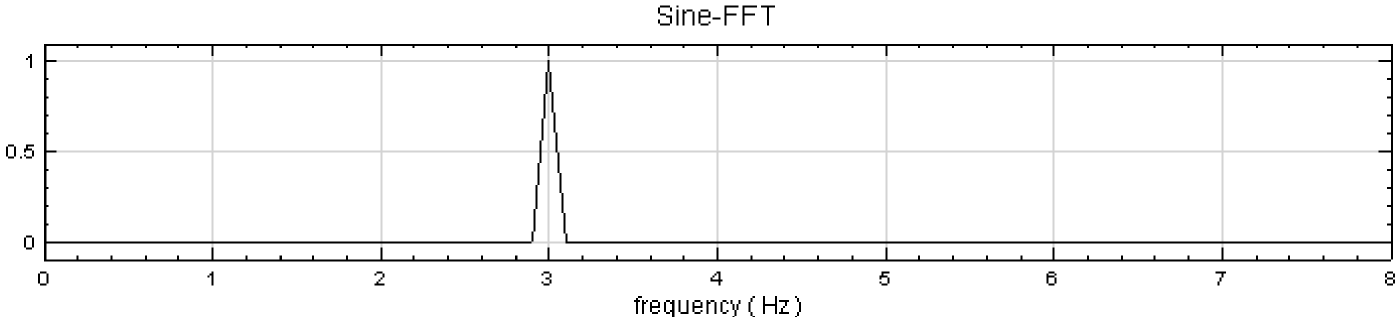
Iterative Gaussian Filter is used to efficiently remove aperiodic components from an input signal.

Introduction

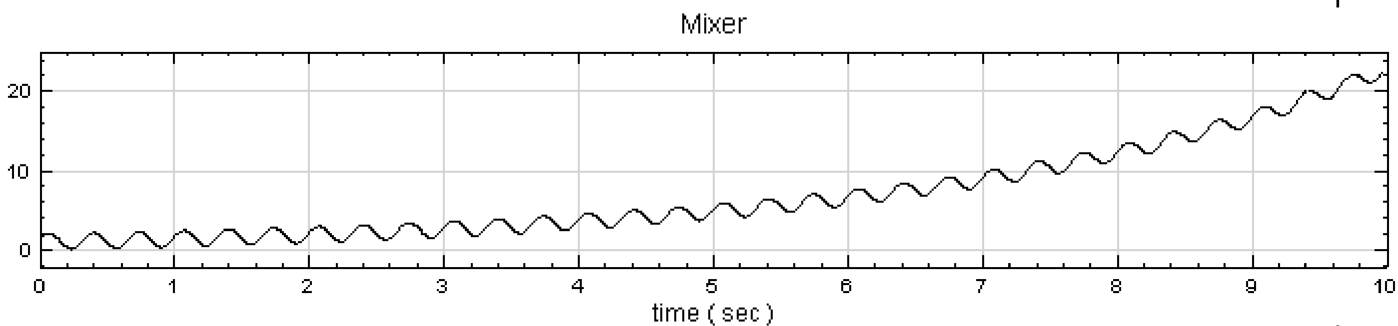
Due to finite observation, there usually exists an aperiodic signal in the data. If the data is processed directly, e.g. using FFT, the resulting spectrum could be wrong. Here is a typical example:

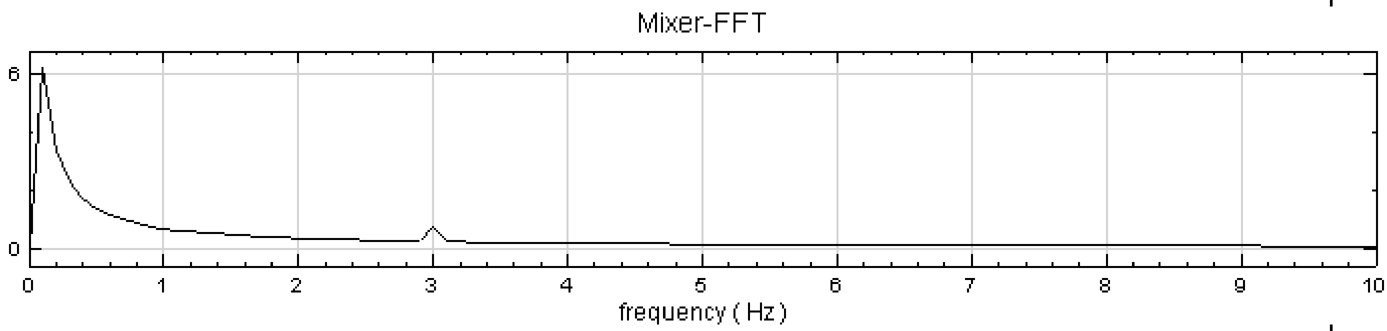
The input data is a Sine Wave. FFT gives a single peak spectrum.





However, if there exists an aperiodic signal emedded in the Sine wave, FFT gives two peak of this data. The power at the very low frequency could be larger than the characteric frequency of the Sine wave.





If simply remove the low frequency components from the data, the spectrum gives the wrong result. And the inverse FFT is not correct either. There is a better way to achieve this.

Assume the data can be represented as the sum of a periodic signal and aperiodic signal,

$$x(t) = \sum_{k=1}^{n} a_k cos\omega_k t + b_k sin\omega_k t + \sum_{i=1}^{n} a_i t^i$$

Follow these steps to filter the data:

Apply FFT to get the spetrum of the input data

Determine F_L 、 F_H and *b* (Attenuation Factor), and calculate two parameters of *Iterative Gaussian Filter* : Smoothing Factor σ and Iteration Number *m.*

First calculate two over parameter k_1 and k_2

$$k_1 = \frac{\ln(1-b)}{\ln(b)}$$

$$k_2 = \left[\frac{F_H}{F_L}\right]^2$$

Use iteration method to solve this equation and determine P_c value

$$0 = P_c + \left(1 - exp(k_1 \times \ln(1 - exp(-\frac{P_c}{k_2})))\right)$$

From P_c value to calculate Gaussian Smoothing Factor (σ) and Iteration Number (m)

$$m = \frac{\ln(1-b)}{\ln(1-e^{-P_c})} + 1$$

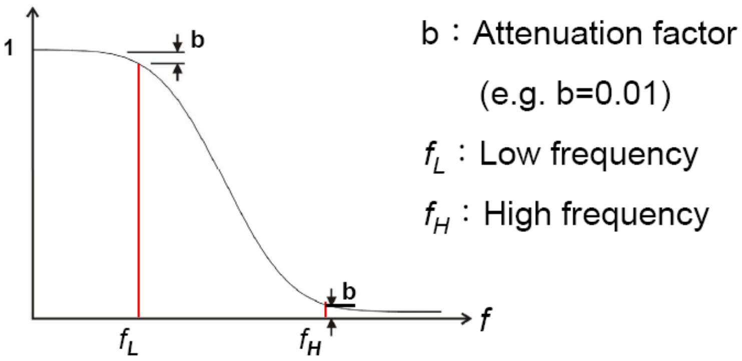
$$\sigma = \frac{1}{\pi}\left(\frac{P_c}{2}\right)^{\frac{1}{2}}$$

Then multify the FFT spectrum with this factor

$$f(\sigma, i, m) = e^{-\frac{t_i^2}{2\sigma^2}}, \; t_i = \frac{i-N}{Sampling\ Frequency}$$

and repeat this filtering m times. So the frequency below F_L and the frequency above F_H are completely filtered out, and the distribution between F_L and F_H is Gaussian.

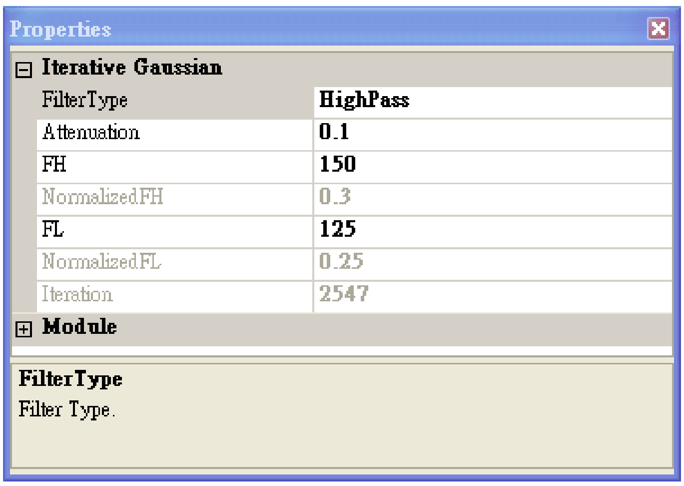
Use Inverse FFT to get the filtered input data.



For LowPass, the frequency below F_L is filtered out. For HighPass, the frequency above F_H is filtered out. The frequency in the middle shows Gaussian distribution; the distribution is determined by b value.

Properties

This module accepts input of Signal which could be real number, single channel or multi-channel, Regular and Audio. The formats of input signal and output signal are identical.

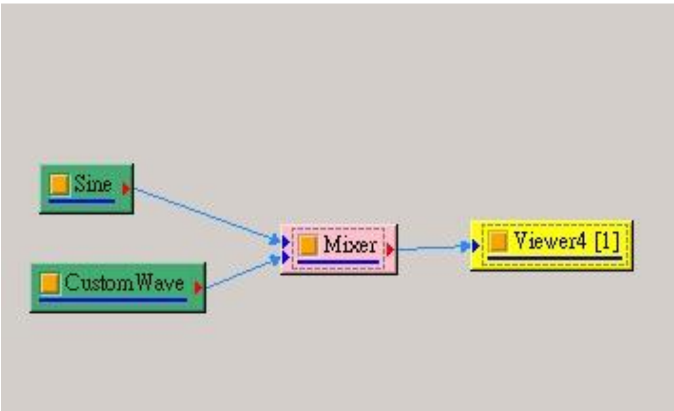


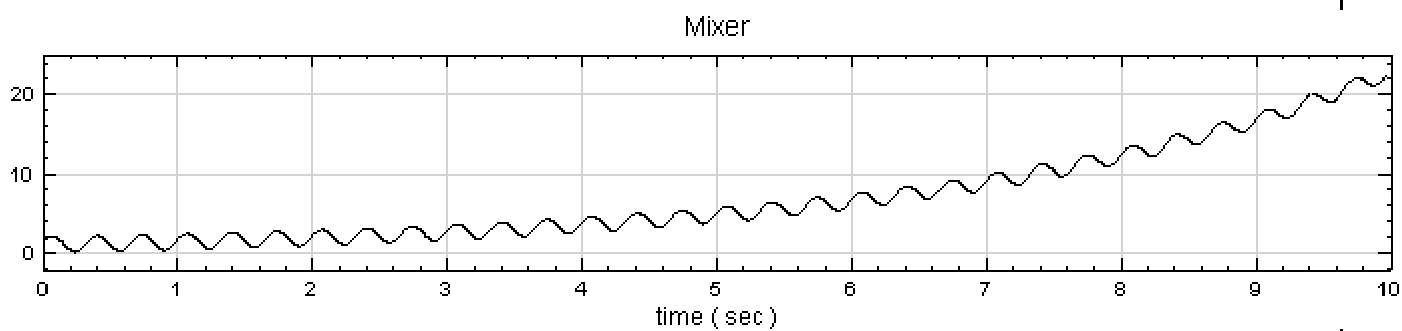
Property Name	Property Definition	Default Value
Filter Type	LowPass: low frequency can pass HighPass: high frequency can pass ByPass: frequency between F_L and F_H can pass	LowPass
Attenuation	The parameter for the Gaussian curve in the filtering	0.01
FH	The high frequency value of the filter	10
FL	The low frequency value of the filter	2

Example

Create a input data in the form of $e^{at}+sin(\omega t)$, then use *Iterative Gauss Filter* to filter out the e^{at} component.

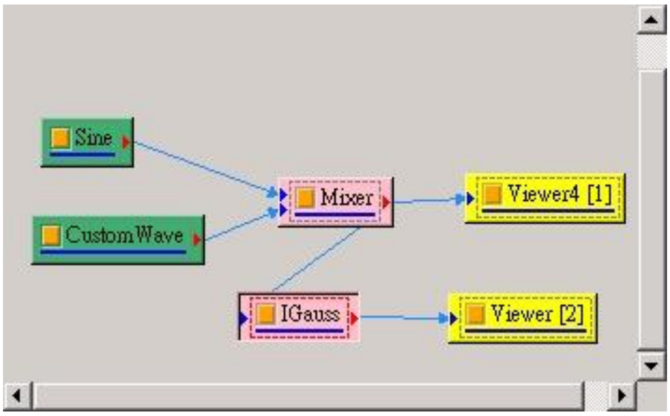
Create a Sine Wave, set TimeLength to 10 seconds and the frequency to 3Hz. And create a *Custom Wave* with setting the *expression* value to be **exp(t/3.2)** and set TimeLength to 10s. Then mix these two signals together and display it using *Channel Viewer*.

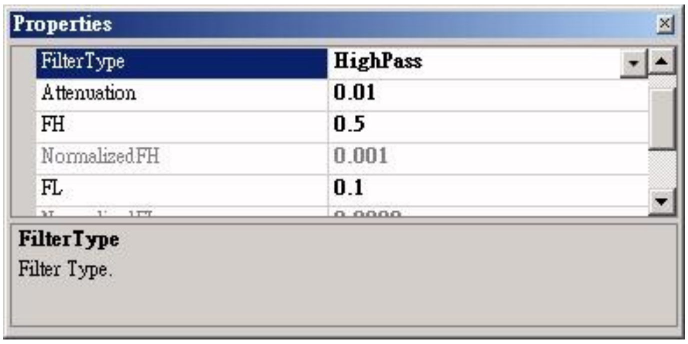




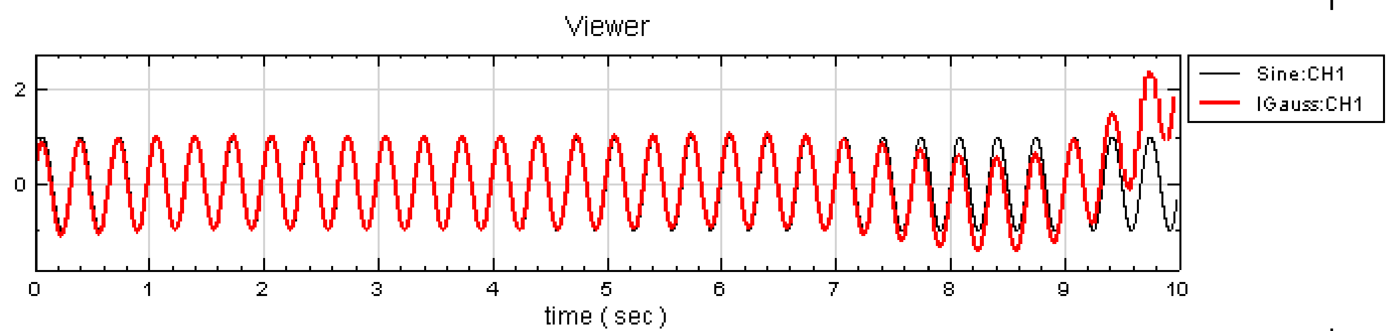
(Try to use FFT to observe its characteristics)

Connect *Mixer* SFO to *Iterative Gaussian Filter*, set Filter Type to highPass, set FH to 0.5, and set FL to 0.1 (Since the Sine wave is at 3Hz, it will pass through the filter with FH is below 3Hz).



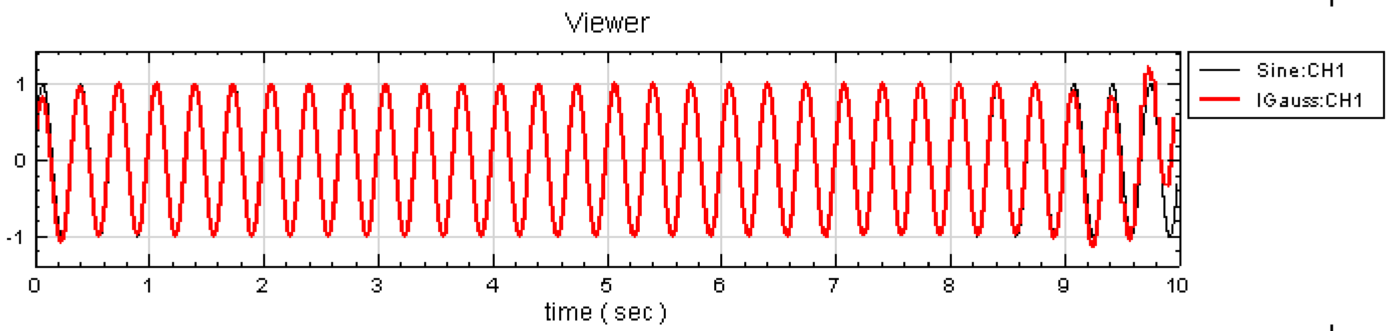


Connect the filtering result to *Channel Viewer*. The result is close to the original signal, except the end part. Adjusting the parameters of the filter can improve the filtering result.



(Thin Black curve: the original Sine Wave; thick red curve: the signal after filtering)

Change *Iterative Gaussian Filter* parameters FH to 2.5 and FL to 0.01, recalculate. Now the filtered signal is very close to the original Sine wave.



Related Functions

Trend Estimator, CustomWave, Fourier Transform.

Reference

1. Diffusive and Fast Filter Using Iterative Gaussian Smoothing, Yih Nen Jeng, Department of Aeronautics and Astronautics, National Cheng Kung University

2. http://www.ancad.com/blog/AnCADSupport/wp-content/uploads/2008/05/it-gauss-2008-7.pdf

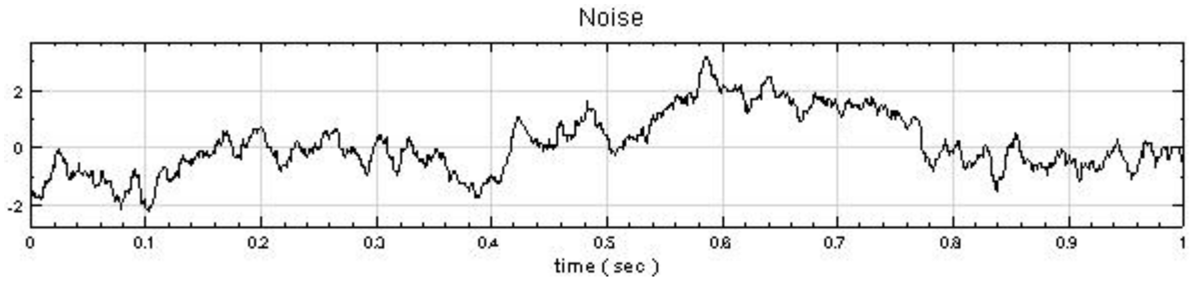
3.2.5 Trend Estimator (Professional Only)

Trend Estimator is a simplified version of *Iterative Gaussian Filter*. *Iterative Gaussian Filter* is explained in another section. If the characteristic of the signal is not fully understood, *Trend Estimator* can be used to estimate its trend without setting parameters as does in *Interative Gaussian Filter*.

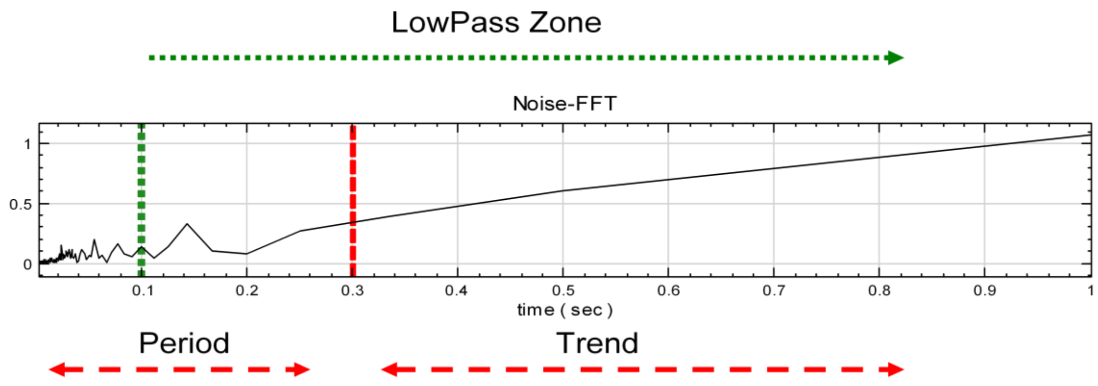
Introduction

The algorithm is the same as the one described in the section of *Iterative Gaussian Filter*. For the parameters of *Trend Estimator*, TrendPeriod and TrendFrequency determine both FL (= 2 / TrendPeriod, 2 * TrendFrequency) and FH (= 4 / TrendPeriod, 4 * TrendFrequency) in *Iterative Gaussian Filter*; then calculation follows the algorithm.

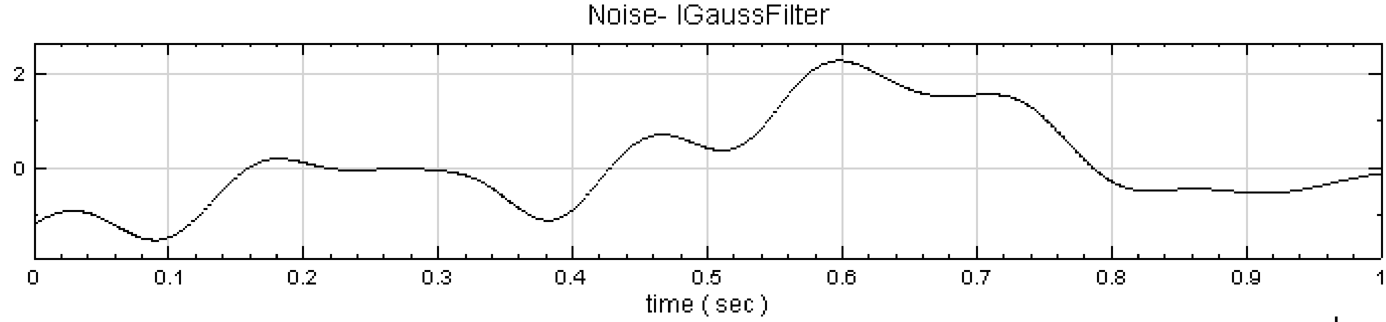
Original signal:



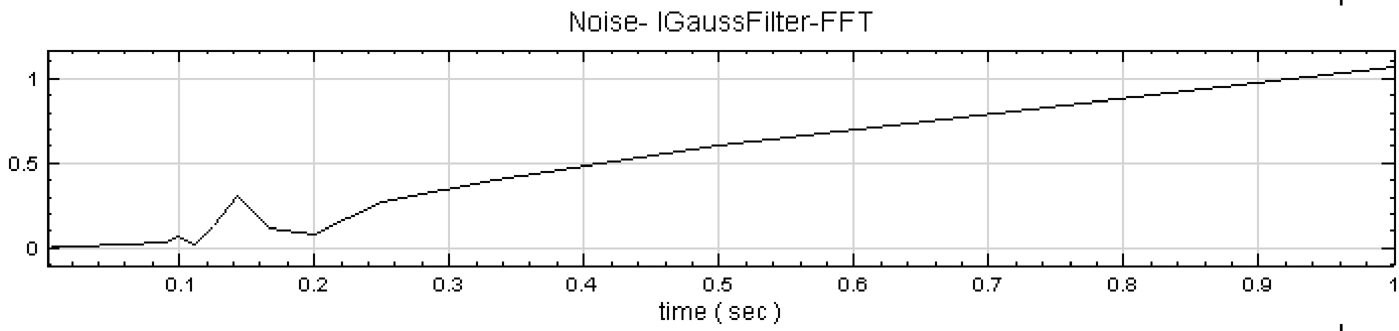
Apply FFT on the signal and the period is set as X axis:



Signal after the low-pass filter, i.e. the trend passes through and the oscillations are filtered out.



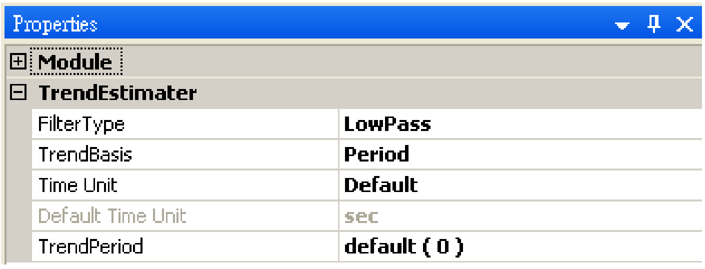
FFT spectrum after filtering, X axis is the period:



It shows that the low periods (high frequency oscillation) have been filtered out.

Properties

This module accepts input of Signal which could be real number, single channel or multi-channel, regular and Audio. The input signal format and the output signal format are identical.



Property Name	Property Definition	Default Value
Filter Type	LowPass: low frequency can pass HighPass: high frequency can pass ByPass: frequency between F_L and F_H can pass	LowPass
Trend Basis	Set Period or Frequency as the reference parameter	Frequency

If Trend Basis set to Period, the parameters are as following,

Property Name	Property Definition	Default Value
Trend Period	If the period is higher than this value, then it is considered as trend. Corresponding parameters	0

	in *Iterative Gaussian Filter*: FL (= 2 /TrendPeriod) FH (= 4 / TrendPeriod)	
Time Unit	Set the unit for TrendPeriod	Frequency
Default Time Unit	The time unit for the input signal	sec

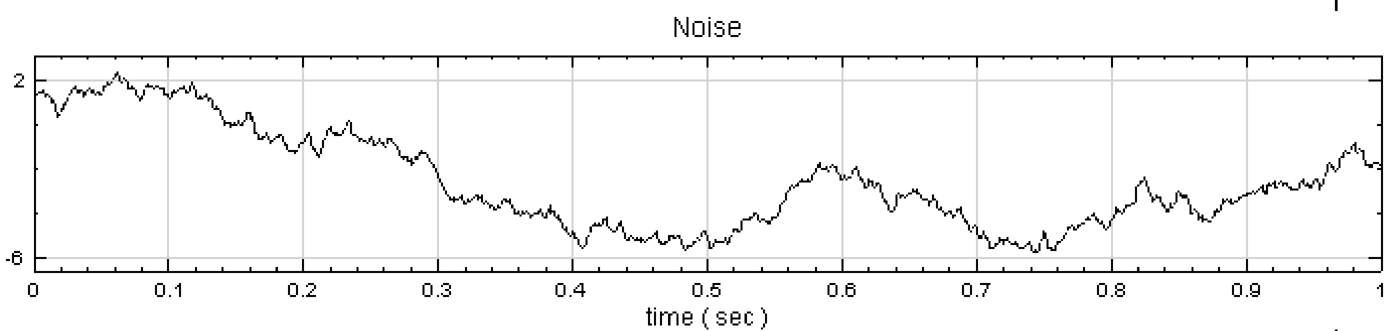
If Trend Basis set to Frequency, the parameters are as following,

Property Name	Property Definition	Default Value
Trend Frequency	If the frequency is lower than this value, then it is considered as trend. Corresponding parameters in *Iterative Gaussian Filter*: FL (= 2 * TrendFrequency) FH (= 4 * TrendFrequency)	0
Frequency Unit	Set the unit for TrendFrequency	Frequency
Default Frequency Unit	The original unit of the FFT on input signal	Hz

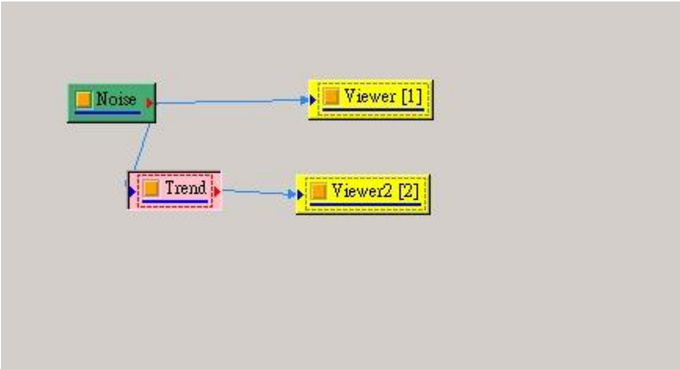
Example

Use *Trend Estimator* to process trend in noise signal.

Create *Source→Noise* signal and set NoiseType in Properties to Brown. The signal looks like:

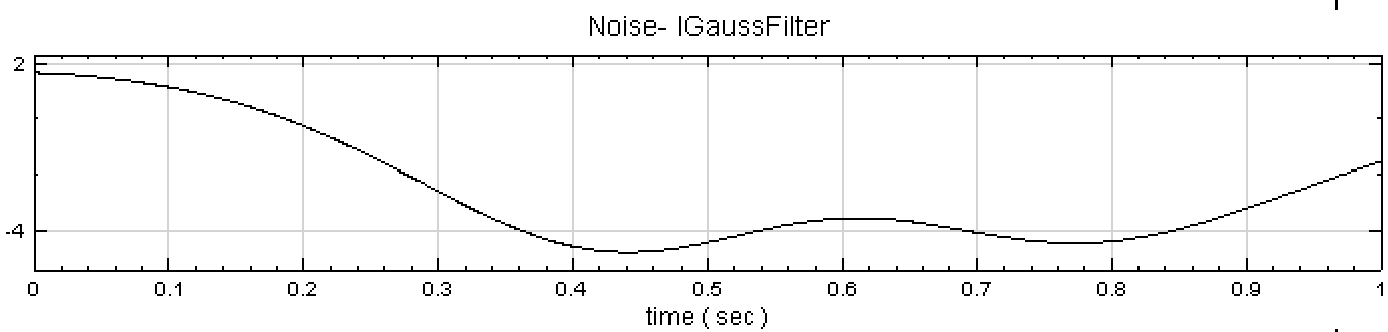


Connect Noise SFO to Trend Edtimator and set TrendPeriod in Properties to 1.0.

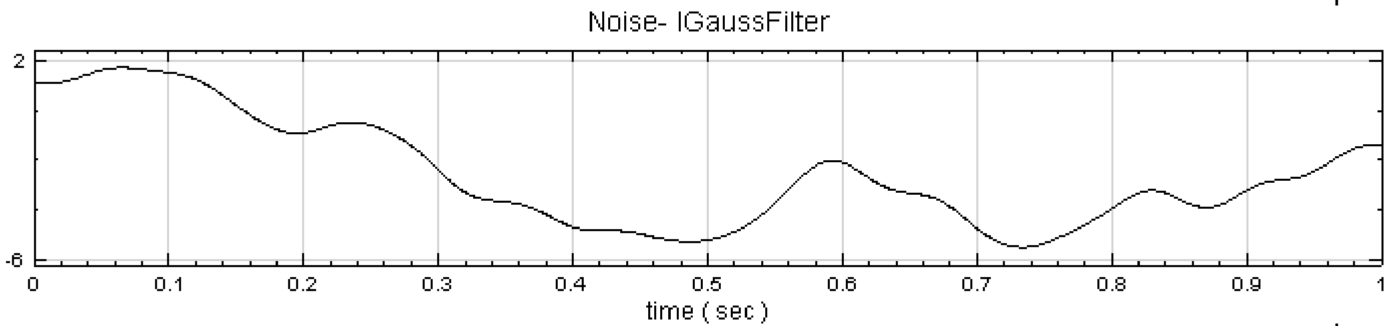


⊞ **Module**	
⊟ **TrendEstimater**	
FilterType	**LowPass**
Time Unit	**sec**
TrendPeriod	**1.0**

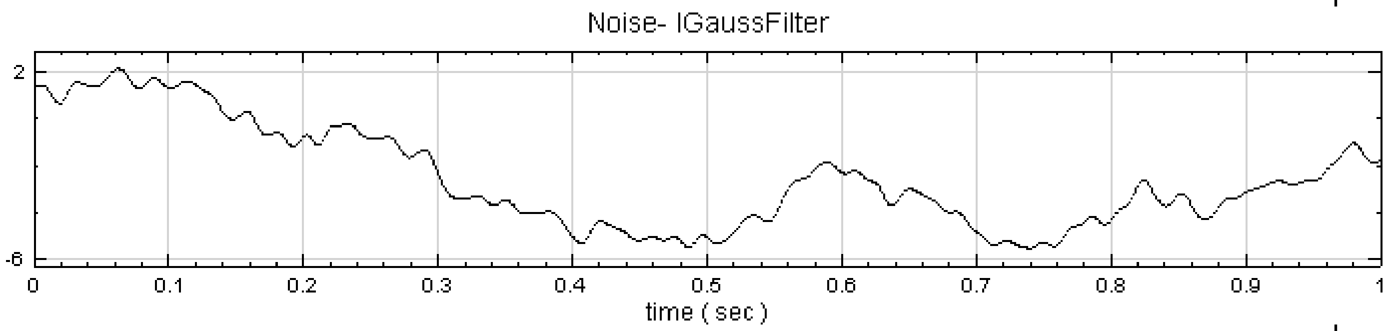
The output shows below:



Set Trend Period to 0.2, the output looks like:



Set Trend Period to 0.05, the output looks like:



It shows that the setting of Trend Period becomes small, signal shows more details. Because Trend Estimator set the lower bound of the period to pass through, the lower the value, the higer frequency (smaller period) to pass through.

Related Instructions

Iterative Gaussian Filter.

Reference

1. Diffusive and Fast Filter Using Iterative Gaussian Smoothing, Yih Nen Jeng, Department of Aeronautics and Astronautics, National Cheng Kung University

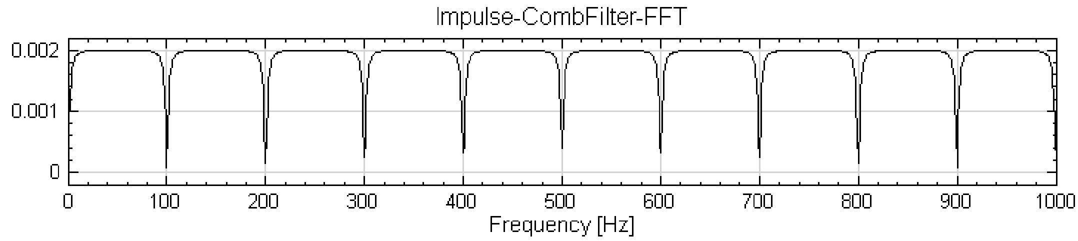
2. http://www.ancad.com/blog/AnCADSupport/wp-connent/uploads/2008/05/it-gauss-2008-7.pdf

3.2.6 Comb Filter

Comb filter can be used to remove or retain a series of frequencies with equal interval.

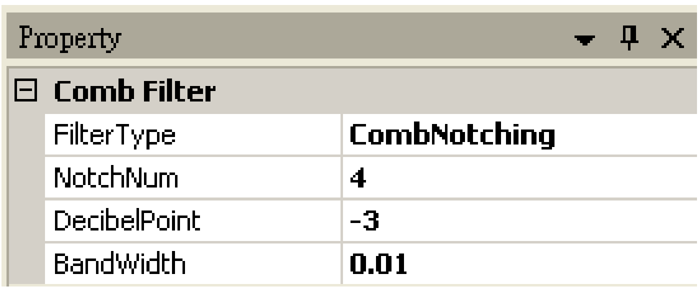
Introduction

The Frequency Response Function of Comb Filter is shown below.



Properties

This module accepts real number, single channel or multi channel, regular signal or audio signal as input signal. The formats of input signal and output signal are identical.

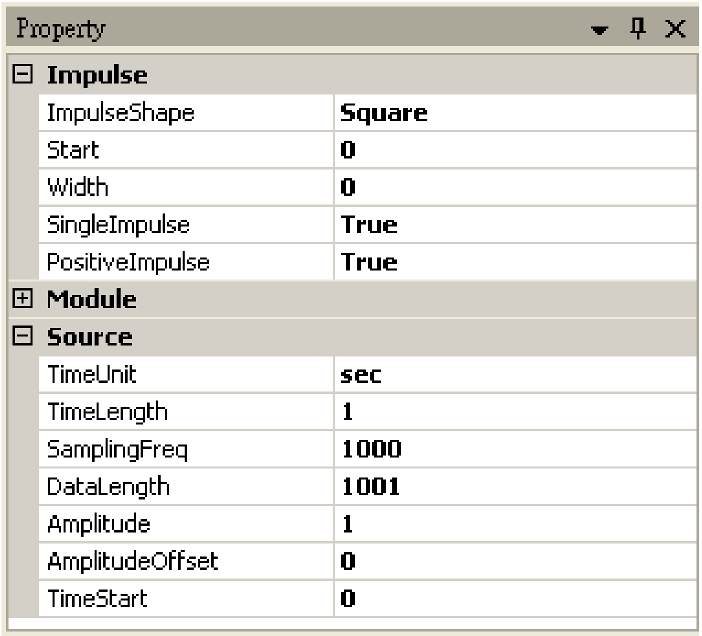


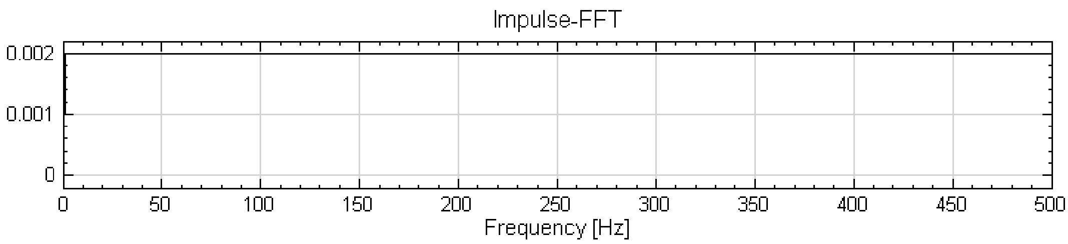
Property Name	Property Definition	Default Value
Filter Type	Filter types consist of CombNotching and CombPeaking	CombNotching
NotchNum	Set the number of notch, which is between 0 and sampling frequency. If the filter type is set as CombNotching, the actual notch number =NotchNum-1.	4
DecibelPoint	Set the decibel point. The smaller value we set, the sharper notch we will get.	-3
BandWidth	Specify the range of frequencies which is	0.01

	specified at a level of ‘DecibelPoint’.The unit is”pi *radians per sample”.	
PhaseCorrection	The phase correction function is enabled when the PhaseCorrection is set to true, vice versa.	True

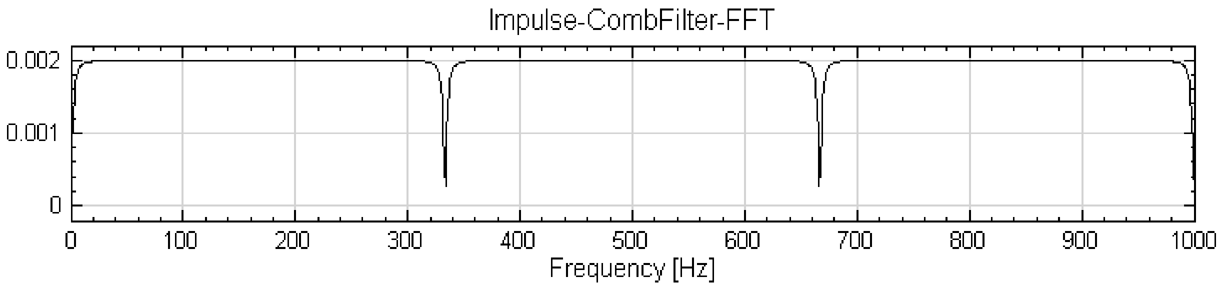
Example

1. Build an Impulse source, Impulse Shape is set as Square，and SingleImpulse is set as True. Then, connect Computet / Transform / Fourier Transform. Lastly, the distribution of frequency is shown by Channel Viewer.



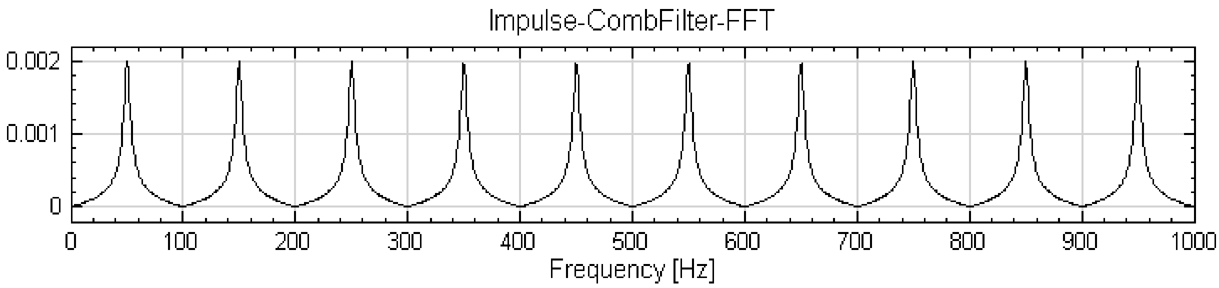


2. Connect Impulse Source with Compute/Filter/CombFilter. Connect Comb with Compute/Transform/Fourier Transform. FFT's Max value is set to 1000, which is equal to sampling frequency. The frequency is filtered with equal interval, when it is viewed by Channel Viewer.



3. Change the Filter Type of Comb Filter as CombPeaking , and NotchNum of it as 11. The result is going to be a series of frequency. The rest of them are going to be removed.

Property	
Comb Filter	
FilterType	**CombPeaking**
NotchNum	**11**
DecibelPoint	**-3**
BandWidth	**0.01**



Related Functions

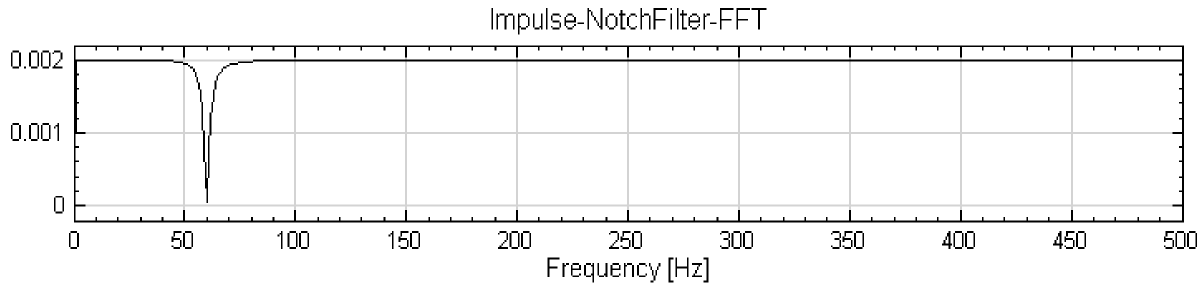
Notch Filter

3.2.7 Notch Filter

Notch Filter is band-reject filter，used to remove a specific frequency。

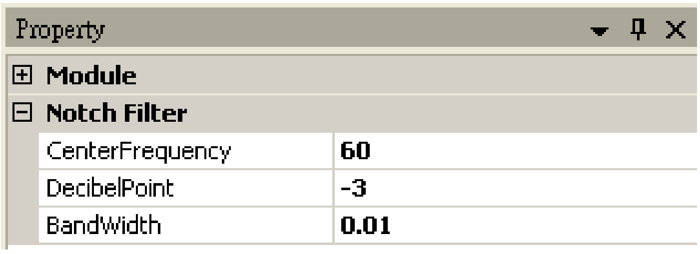
Introduction

Notch Filter is used to remove a specific frequency. For example, the frequency we want to remove is 60 Hz. The frequency response function is shown as follows.



Properties

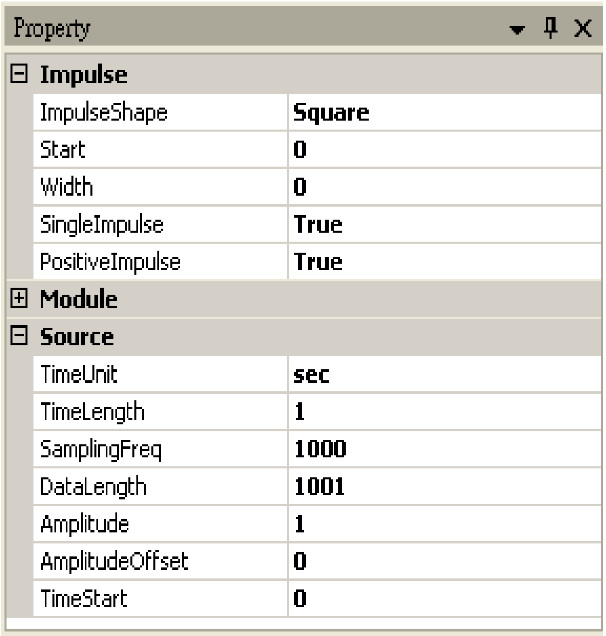
This module accepts all kinds of inputs of signal, such as real number, single channel multi-channel, regular signal or audio signal. The formats of input signal and output signal are identical. The definition of property is shown as follows.



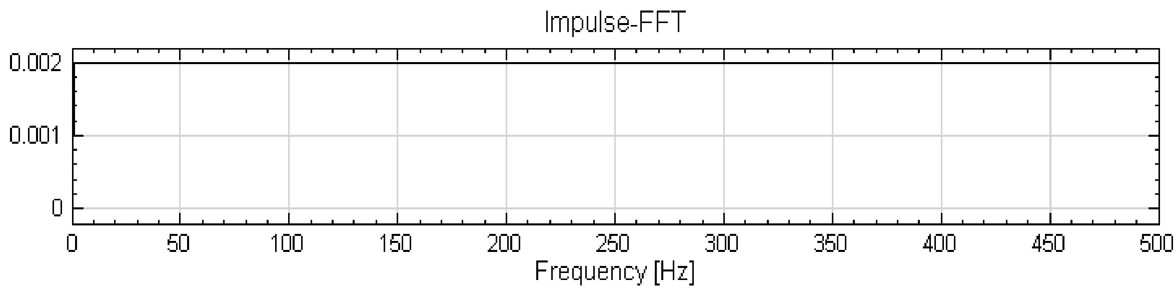
Property Name	Property Definition	Default Value
CenterFrequency	The center frequency which is supposed to be removed.	60
DecibelPoint	The smaller value we set, the sharper notch we will get.	-3
BandWidth	Specify the range of frequencies which is specified at a level of ‘DecibelPoint’.The unit is”pi *radians per sample”.	0.01

Example

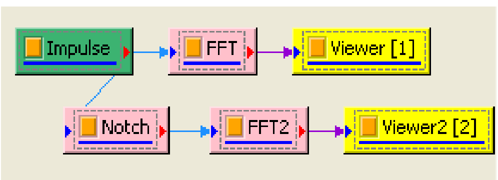
1. Build an Impulse source, Impulse Shape is set as Square, and SingleImpulse is set as True. Then, connect Compute / Transform / Fourier Transform. Lastly, the distribution of frequency is shown by Channel Viewer.

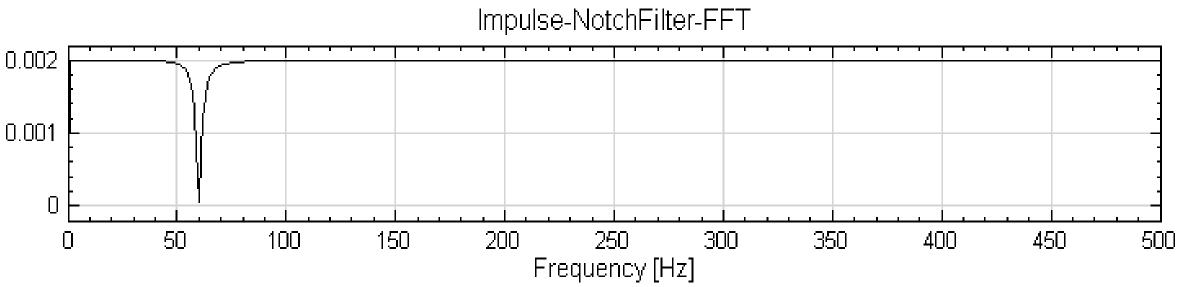


Property	
Impulse	
ImpulseShape	Square
Start	0
Width	0
SingleImpulse	True
PositiveImpulse	True
Module	
Source	
TimeUnit	sec
TimeLength	1
SamplingFreq	1000
DataLength	1001
Amplitude	1
AmplitudeOffset	0
TimeStart	0

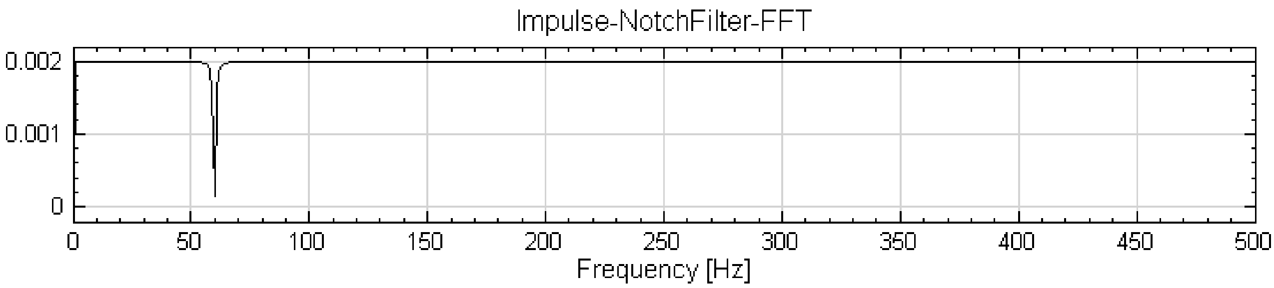


2. Connect Impulse Source with Compute/Filter/CombFilter. Connect Notch with Compute/Transform/Fourier Transform. The result is shown by Channel Viewer. It is can be seen that 60 Hz is removed.





3. Change Notch Filter's DecibelPoint into -10， and BandWidth into 0.001. The removed frequency and range become sharper.



Related Functions

Comb Filter

3.3 Mathematics

This group of modules processes signals or relationship between signals mathematically, whose components are listed below.

1. Remove DC: To remove the direct current component of signal.
2. Mixer: To add (or subtract) several signals using identical time scale.
3. Multiplier: To multiply several signals using identical time scale
4. Math: To input mathematical formula for signal calculation.
5. Diff: To calculate approximate differentiation of input signal.
6. Integrate: To calculate approximate integration of input signal.
7. DoMatlab: To connect signals to Matlab, compile and run the Matlab code.

3.3.1 Remove DC

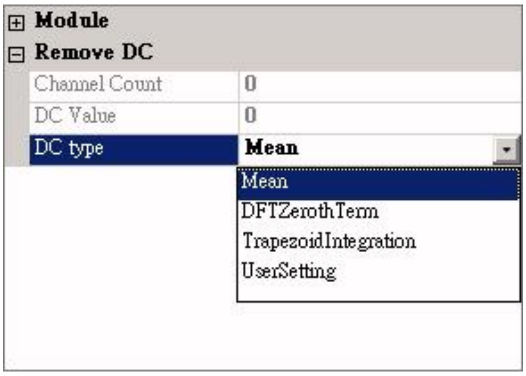
Remove the signal direct current component, i.e. to remove the signal shift along Y-axis.

Introduction

Let the signal source be $X = \{x_0, x_1 ... x_{N-1}\}$, its average is $\bar{x}$, i.e. DC. Hereafter, $X' = \{x_0 - \bar{x}, x_1 - \bar{x} ... x_{N-1} - \bar{x}\}$ is said to be *Remove DC*.

Properties

This module accepts input of Signal (which could be real number, single channel, Regular) and Audio (which could be real number, single channel, Regular). The property is DC type which includes four types of calculation methods to compute the shifting along the Y-axis, where the default option is Mean. The detailed meaning of these methods is given in the table below.



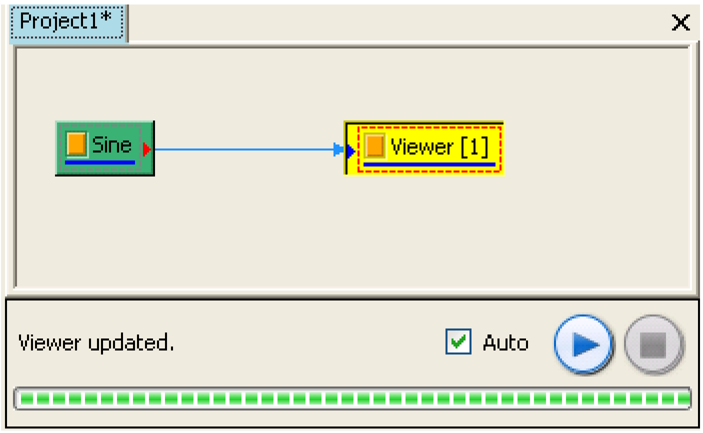
DC type Property	Property Definition
Mean	To calculate arithmetic average
DFT_First_Amplitude	After performing Fourier Transform on the original data, define X-axis as zero point which has the value DC($DC \equiv \frac{a_0}{2} = \frac{1}{2}\int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt\Big\|_{\omega=0}$)
Trapezoid_integration	Divide the result of Trapezoid Integration by the total number of points. The result is the DC.

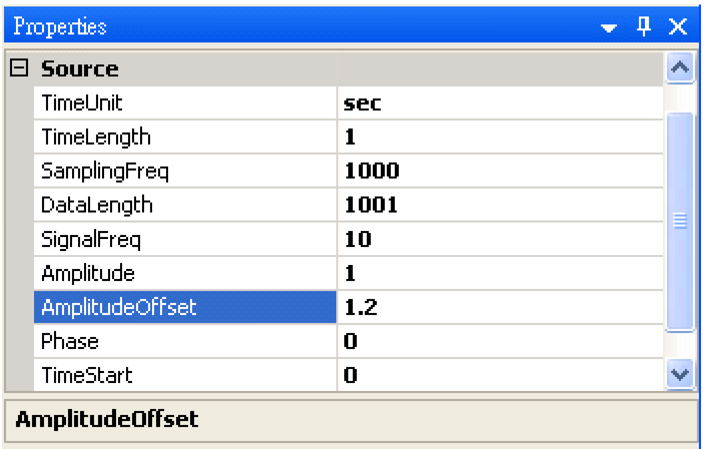
User_Setting	The users can set the desired shift value manually.

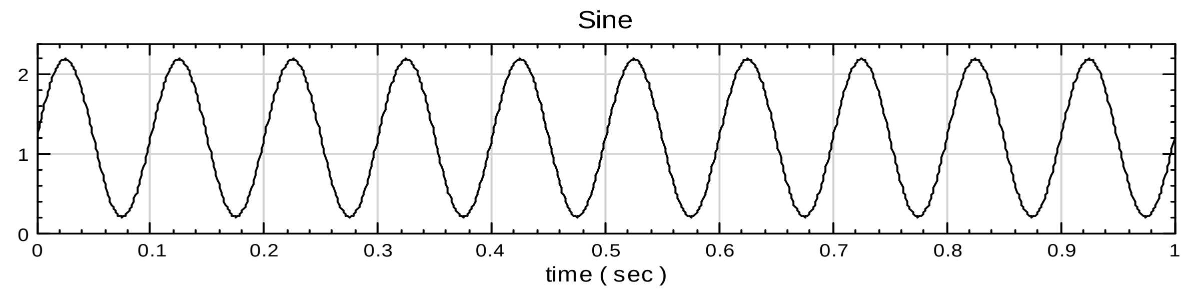
Example

Create a sine wave which is shifted along Y-axis and then use RemoveDC to remove the shift.

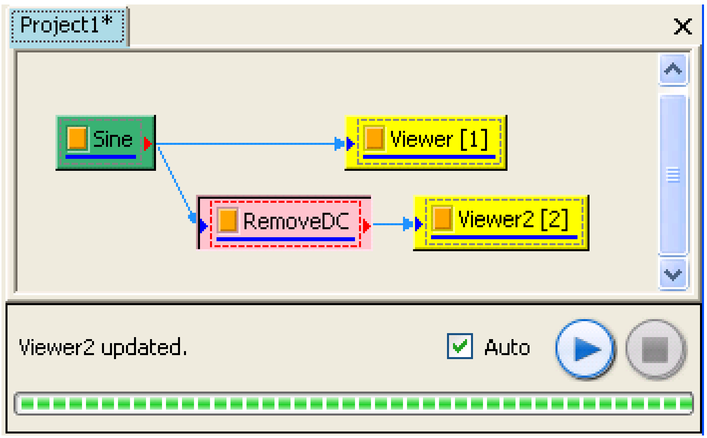
1. Create a sine wave using the *Source*→*Sine Wave*, and then adjust the *Source*→*AmplitudeOffset* to 1.2 to shift the signal along the Y-axis in positive direction for 1.2 unit. Next, use the *Viewer*→*Channel View* to observe.

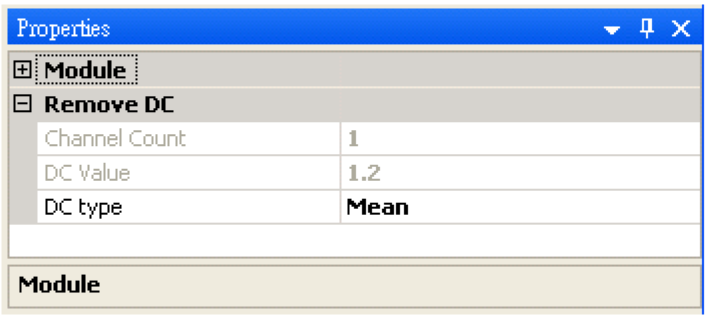


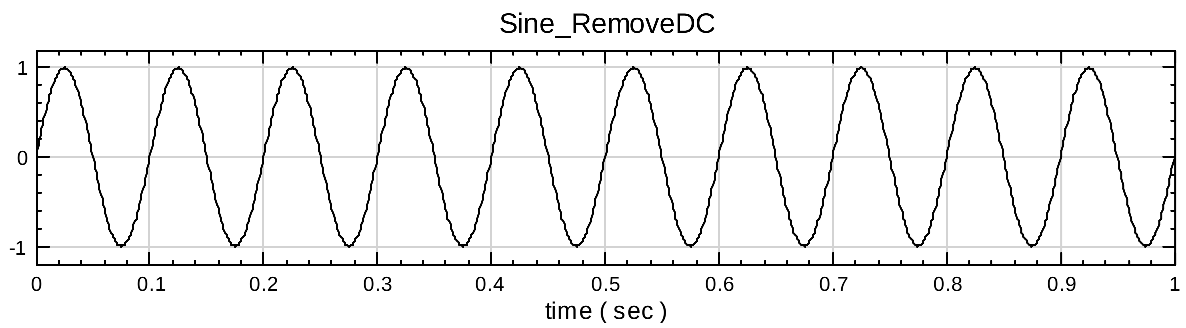




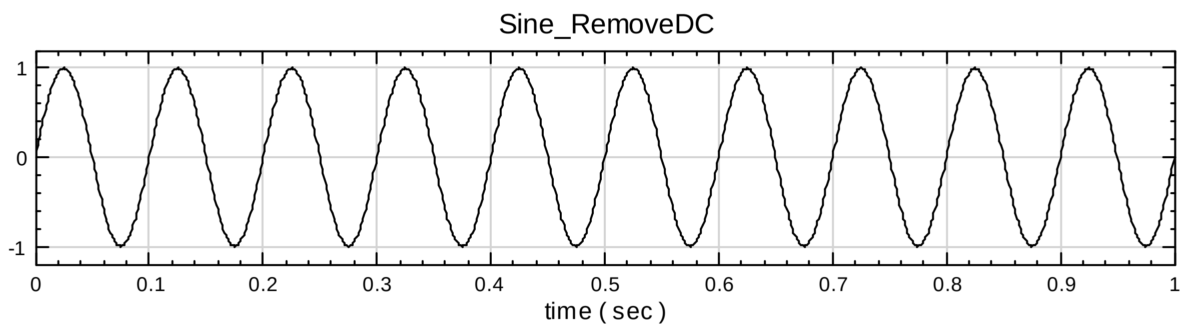
2. Connect the original signal to *Compute→Mathematics→ RemoveDC* and set the method as *Mean* in *DCType*. It can be seen that the shift is removed.



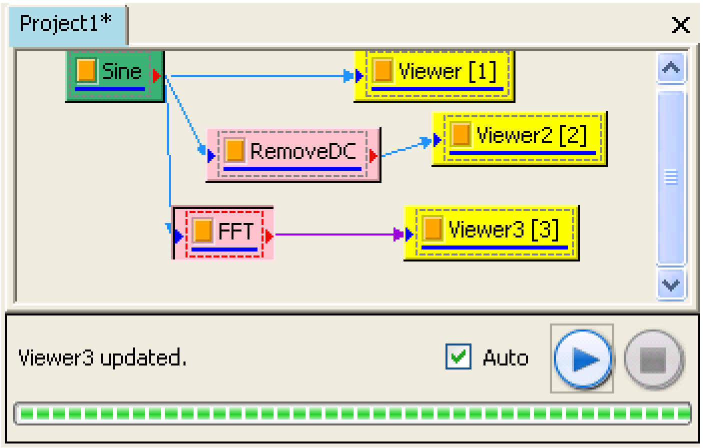


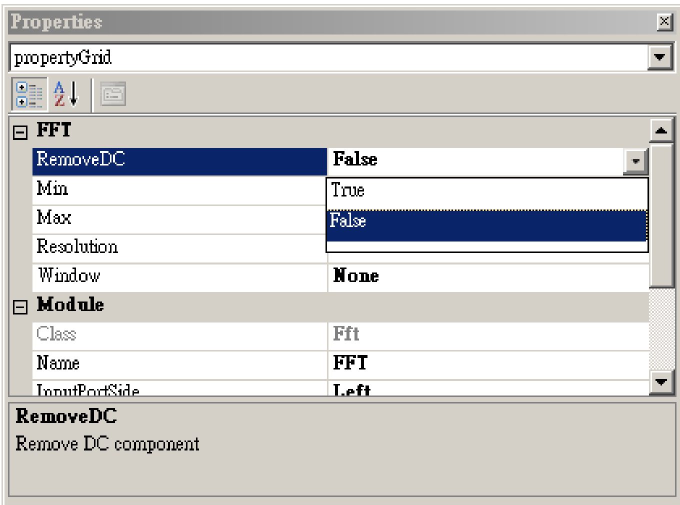


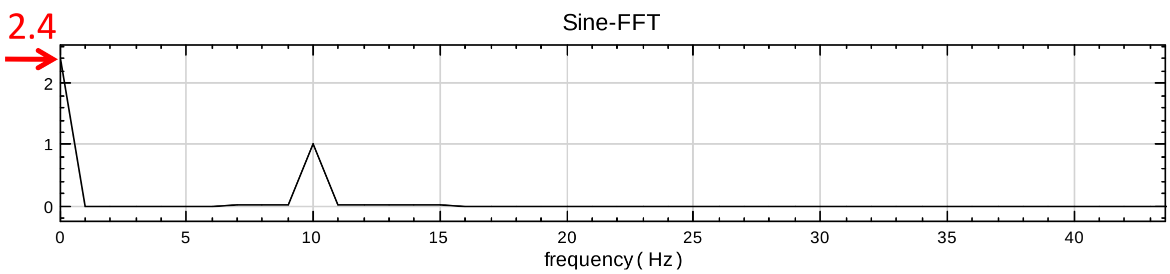
3. The DC Type can be chagned also, e.g. *DFTZerothAmplitude*. However, in this example, the result would be identical.



4. Connect the signal to *Compute*→*Transform*→*Fourier Transform* to perform Fourier Transform. Without the *RemoveDC* in *Fourier Transform*, it can be seen that the amplitude at 0Hz is 2 times of 1.2.







For horizontal shifting along time axis, please reference to Channel→*TimeShift* module.

Related Functions

Source, Fourier Transform, TimeShift.

3.3.2 Mixer

Mixer is used to mix several signals.

Introduction

Assume N groups of signals, $X^{(n)}(t)$, each group of which has different time axis *t* and sampling frequency. The mixed signal $Z(t)$ is

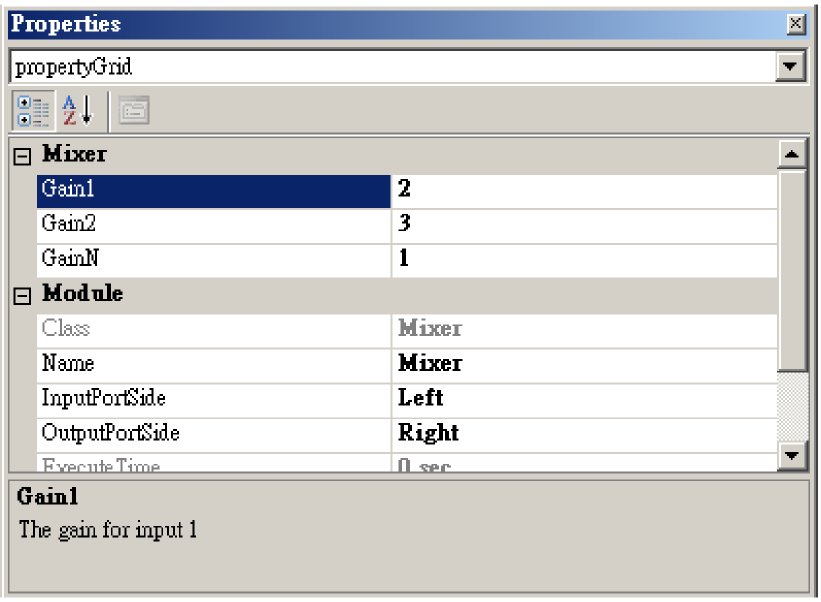
$$Z(t_i) = a \cdot X^{(1)}(t_i) + b \cdot X^{(2)}(t_i) + c \cdot (\sum_{k=3}^{n} X^{(k)}(t_i)) \text{, where } a,\ b,\ c \text{ are weights}$$

In this module, because the time-axis of input signals are supposed to be different, the minimum sampling frequency $freq_{\min}$ of the input signals is extracted first, then all other signals are re-sampled by $freq_{\min}$. After the time-axis of all input signals are unified, the signals are added at every time points. Notice that the weights from the 3rd group of signal are all equal to c.

Properties

This module accepts input of Signal (which could be real number, single channel, Regular) and Audio (which could be real number, single channel, Regular). Multiple signal input is also allowed.

Gain1, Gain2 and GainN are the weights of the first, the second and the third group of input signals, respectively. The difference between this module and Math is that the Mixer can perform faster addition/subtraction computation, and also perform addition/subtraction on signals with different length, while Math does not have this type of functionality.

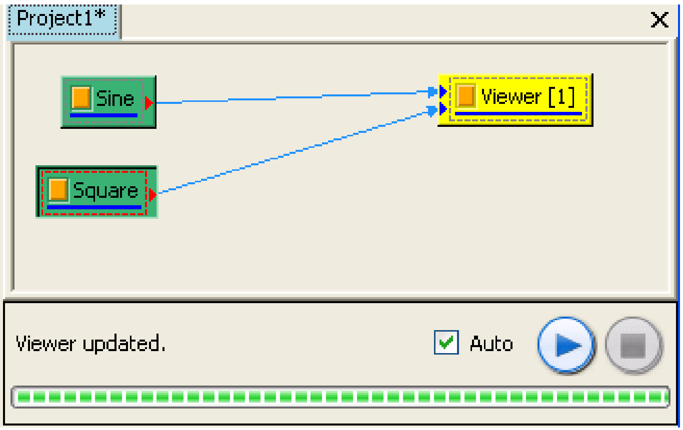


Property Name	Property Definition	Default Value
Gain1	Set the weight a for the first group of signal	$a=1$
Gain2	Set the weight b for the second group of signal	$b=1$
GainN	Set the weight c for the signals from the 3rd group	$c=1$

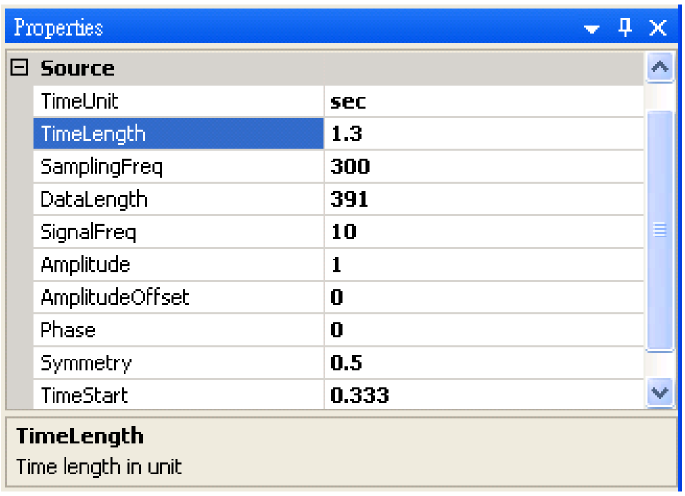
Example

This example below shows the procedure to mix a sine wave and a square wave with different time axis.

1. Use *Source→Sine Wave* to create a signal with frequency of 5Hz, sampling frequency of 1000Hz and duration of 1.5 seconds. And then create a square wave with frequency of 10Hz, sampling frequency of 300 Hz, duration of 1.3 seconds, and time starting point of 0.333 second. Next, use *Viewer→Channel Viewer* to observe the wave.



Sine properties are shown in the table below.

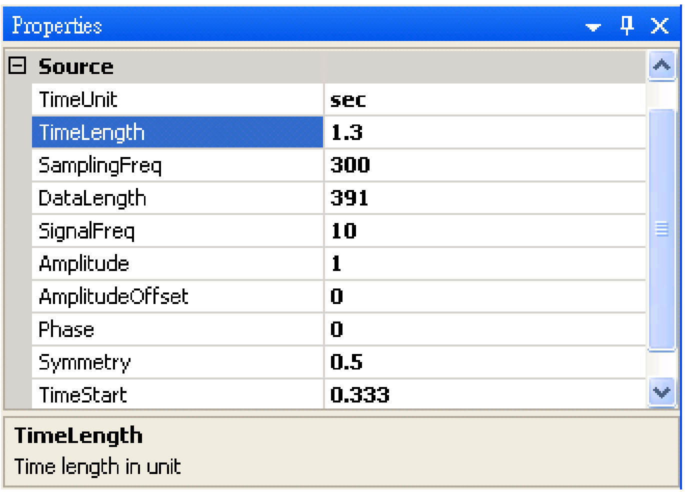


Properties

Source	
TimeUnit	sec
TimeLength	1.3
SamplingFreq	300
DataLength	391
SignalFreq	10
Amplitude	1
AmplitudeOffset	0
Phase	0
Symmetry	0.5
TimeStart	0.333

TimeLength
Time length in unit

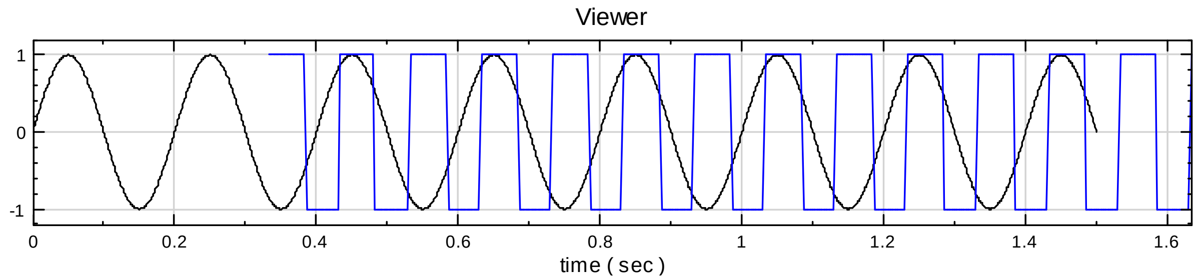
Square properties are shown in the table below.



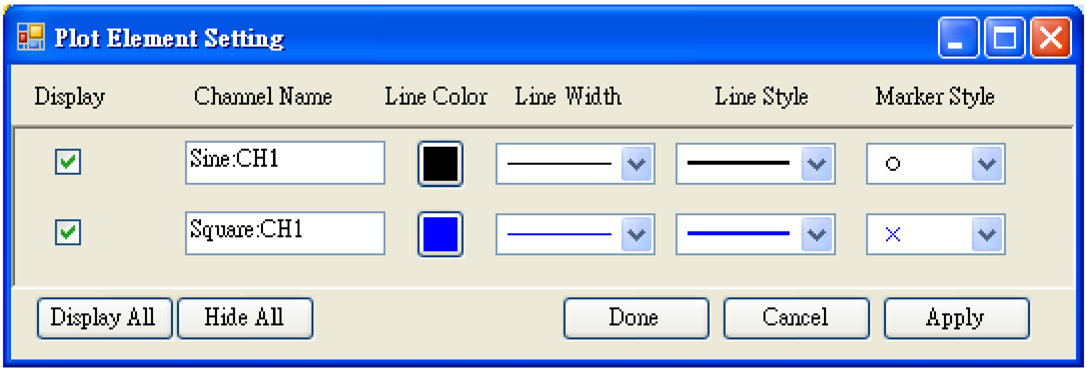
Properties

Source	
TimeUnit	sec
TimeLength	1.3
SamplingFreq	300
DataLength	391
SignalFreq	10
Amplitude	1
AmplitudeOffset	0
Phase	0
Symmetry	0.5
TimeStart	0.333

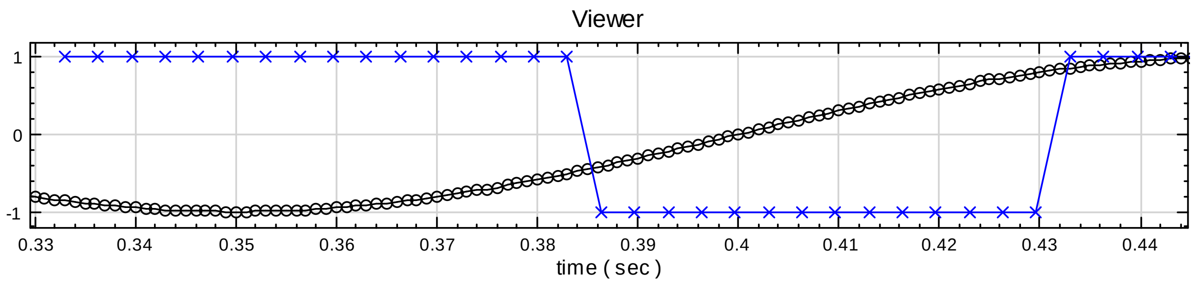
TimeLength
Time length in unit



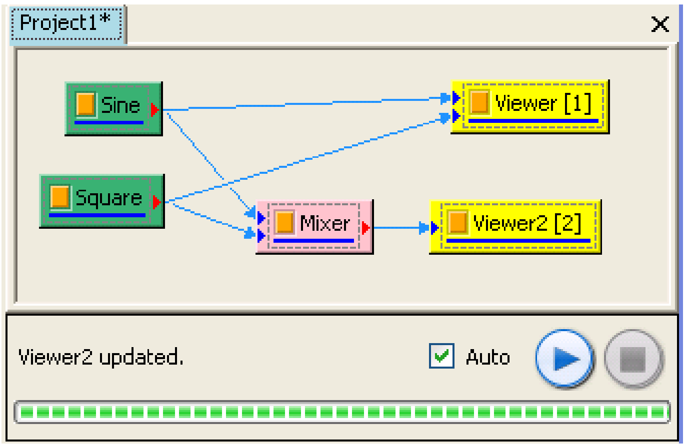
Select *PlotEditor* in the *Properties/Representation→Plot Elem Editor* in *Channel Viewer*. In the popped-up *Plot Element Setting* window, add 「o」 to Sine curve, 「x」 to Square curve, and use tool of *Zoom X* to enlarge the onverlapping point of these two signals. It can be seen that the signal data-point distributions along X-axis (time-axis) are completely different. Settings of *PlotEditor* are given as follows.

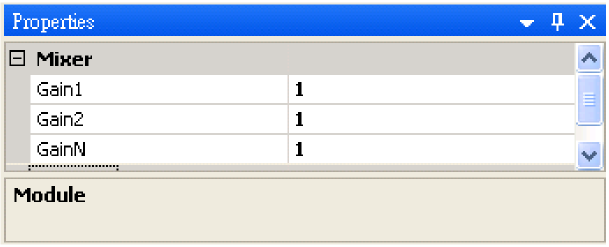


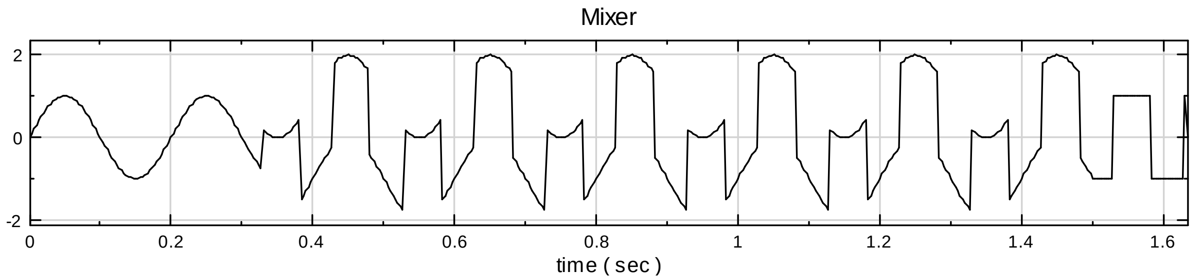
Zoom in the result in *Chanel Viewer* as below.



2. Add these two signals to form a new signal. As shown below in the Network, use *Compute→Mathematics→Mixer* to perform the signal mixture. The first input to *Mixer* is Sine, the second input is Square and the corresponding properties are Gain 1 and Gain 1, respectively. Both have default value of 1. Next, use *Viewer→ Channel Viewer* to plot the *Mixer* wave.

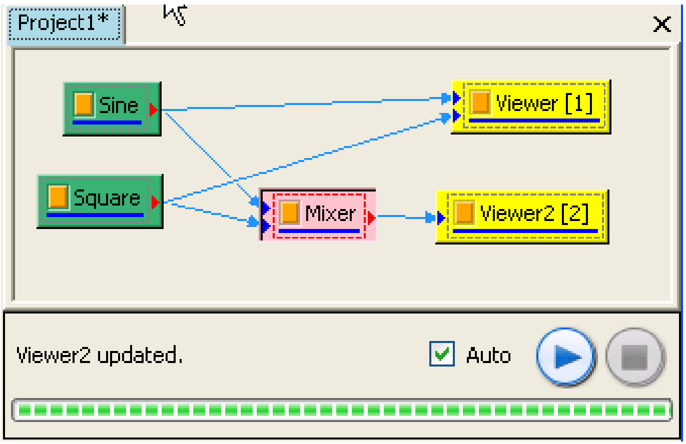


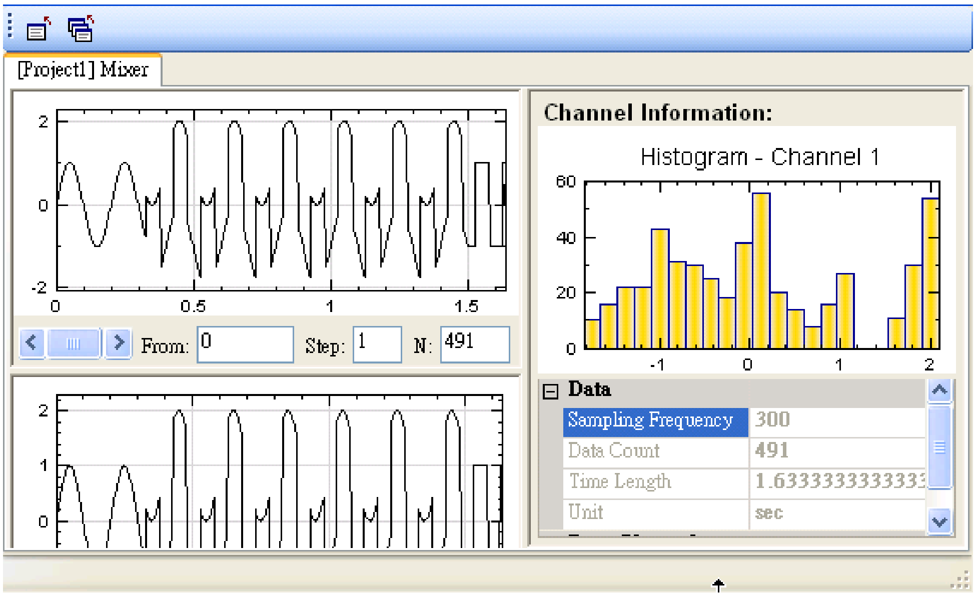




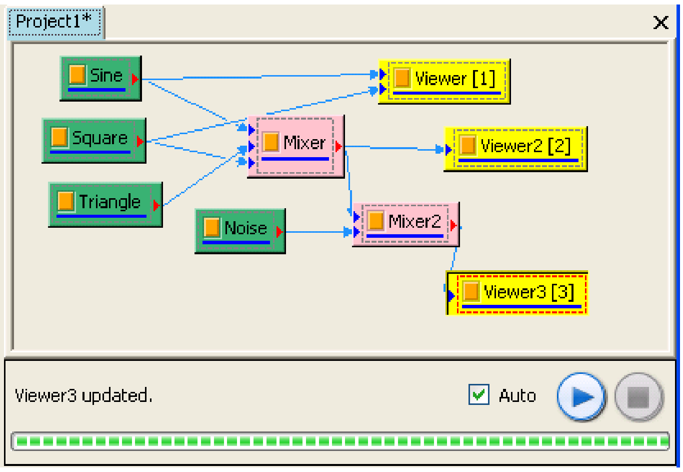
3. Now, use *Data Viewer* to check the sampling frequency and duration of the output signal from *Mixer*. The sampling Frequency is 300 Hz. The signal starts at 0 second and ends at 1.63333 second.

The computation method in *Mixer* uses the duration of mixed input signal as the total duration, selects the minimum sampling frequency from the input signals, and adds all signals after they are multiplied by corresponding weights. Therefore, to use *Mixer*, special attention must be paid to the sampling frequency of input and output signals.





4. Notice that more than 3 groups of data can also be mixed. However, for all data groups which are higher than three, the weights are all set as GainN. Therefore, it is not encouraged to use one Mixer to mix more than 3 groups of data. It is recommended to use multi-layer Mixer to achieve mixture of more than 3 groups of data. The figure below shows an example using this method to mix multiple signals.



Related Functions

Channel switch, Multiplier, Sine, Square.

3.3.3 Multiplier

This component multiplies multiple input signals.

Introduction

Mathematically, assume N groups of signal sources, $X^{(n)}(t)$, where time-axis t and sampling frequency of every signal are not necessarily to be identical. The mixed signal $Z(t)$ is

$$Z(t_i) = X^{(1)}(t_i) \cdot X^{(2)}(t_i) \cdots \cdot X^{(N)}(t_i)$$

In this module, because the time-axis of input signals are supposed to be different, the minimum sampling frequency, $freq_{\min}$, in all input signals is extracted first, and then all other signals are re-sampled by $freq_{\min}$. After the time-axis of all input signals are unified, the signals are multiplied at every time points.

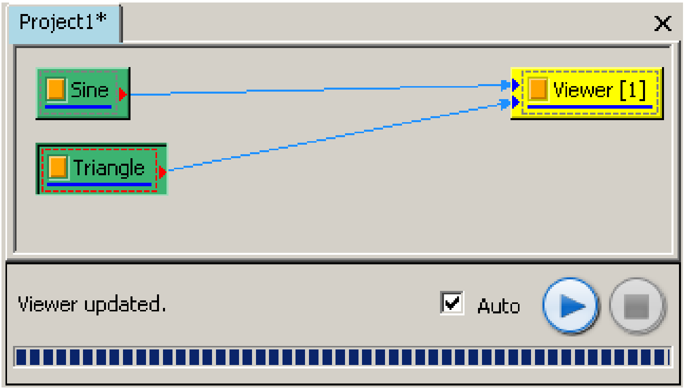
Properties

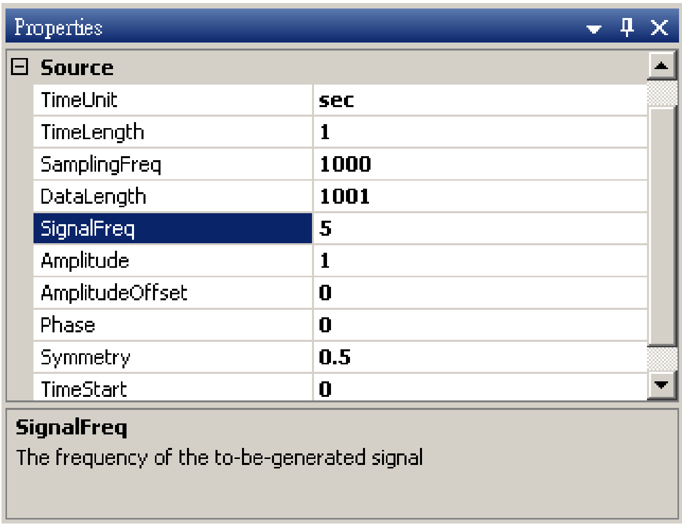
This module accepts input of Signal (which could be real number or complex number, single channel, Regular) and Audio (which could be real number or complex number, single channel, Regular). Multiple signal input is also allowed. This module does not require default values. It can perform multiplication on signals which have different length and sampling frequency.

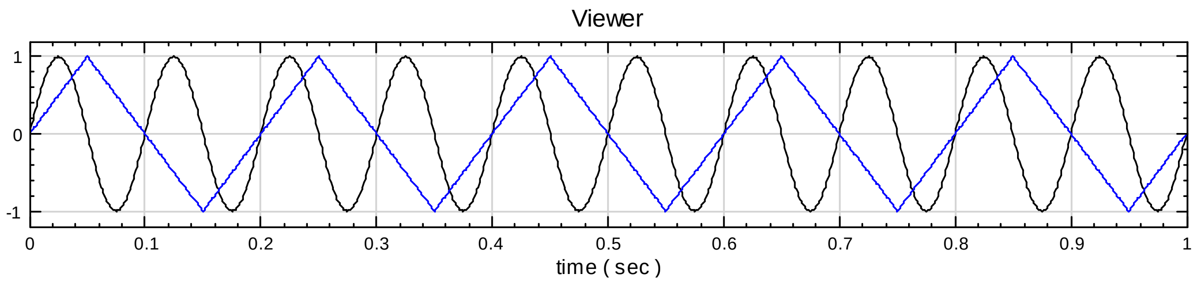
Example

This example shows the multiplication of a sine wave and a triangular wave.

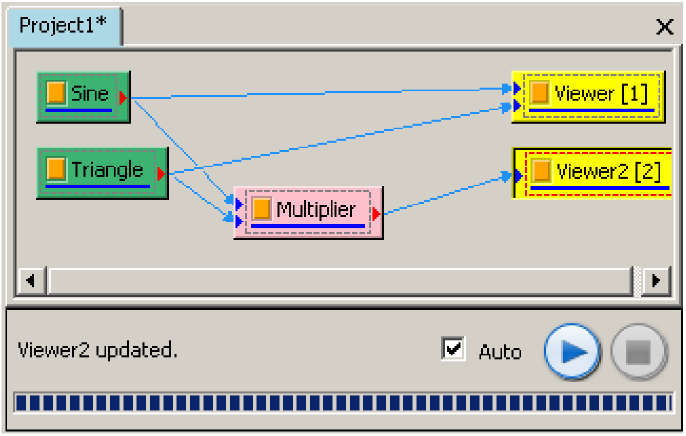
1. Use *Source→Sine Wave* and *Triangle Wave* to generate a sine wave and a triangle wave. Change the *Properties/SignalFreq* of the triangle wave to 5 and use the *Viewer→Channel Viewer* to observe the original wave.

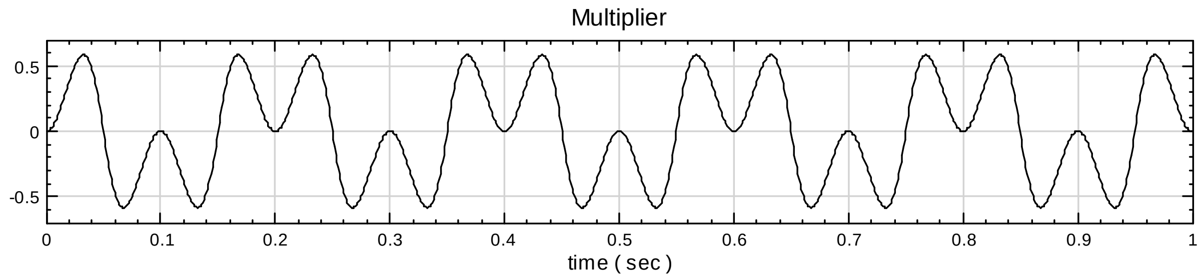




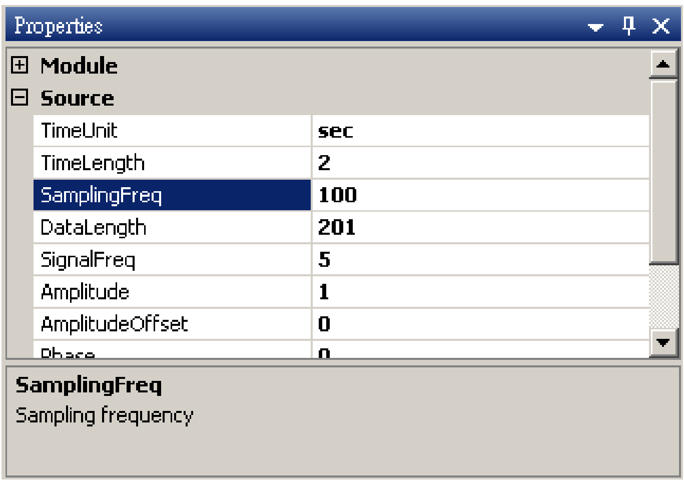


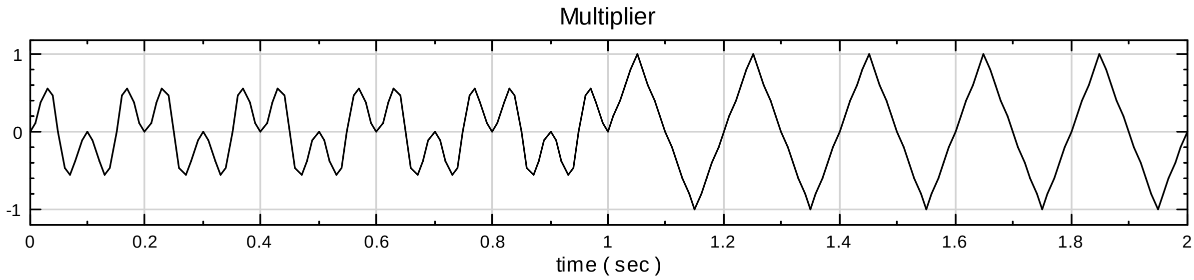
Multiply these two signals using *Compute→Mathematics→ Multiplier*. The output signals are shown as below.





2. As *Mixer*, the *Multiplier* allows the sampling frequency and time length of the two signals to be different. The sampling frequency of the output signal is identical to the minimum sampling frequency in the input ones. On the time axis, the overlapping parts of the input signals are multiplied together while the other part is intact. Change the SignalFreq of the triangular wave to 100, TimeLength to 2. Then, in the output signal, the Signal Frequency would be 100 and the Time length would be 2 seconds.





Related functions

Channel Switch, Mixer, Viewer, Source.

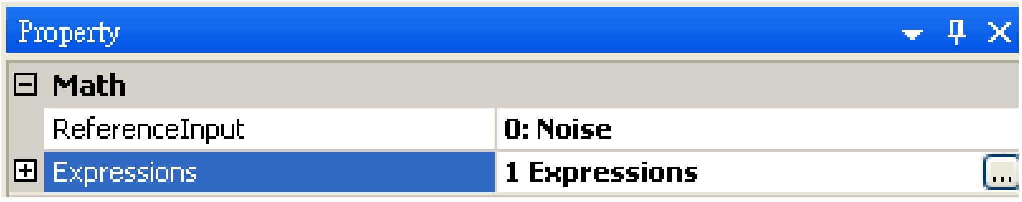
3.3.4 Math (Professional Only)

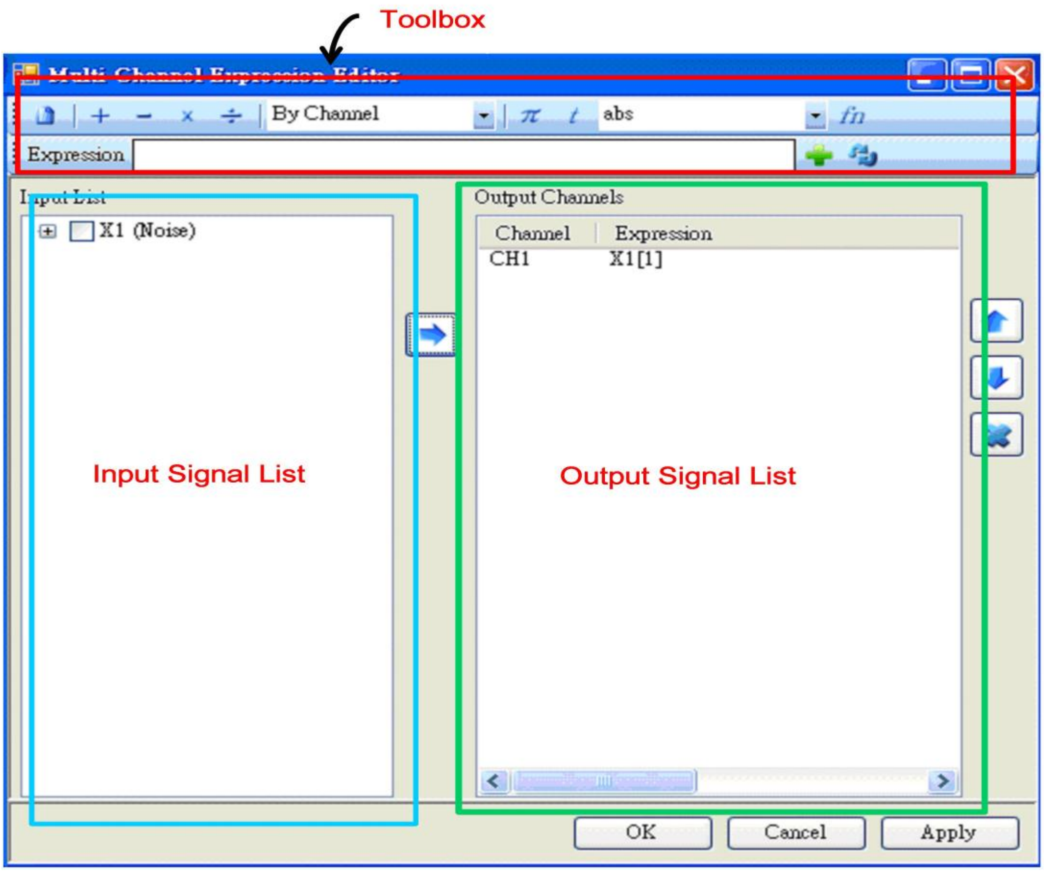
Do point to point math calculation for input signal.

Interface Introduction

This module accepts input of Signal which could be real number or complex number, single channel or multi-channel, regular and audio.

To active the module, click the button  to the right of the field Properties / Expressions. Then MultiChannel Expression Editor is popped up as shown below.

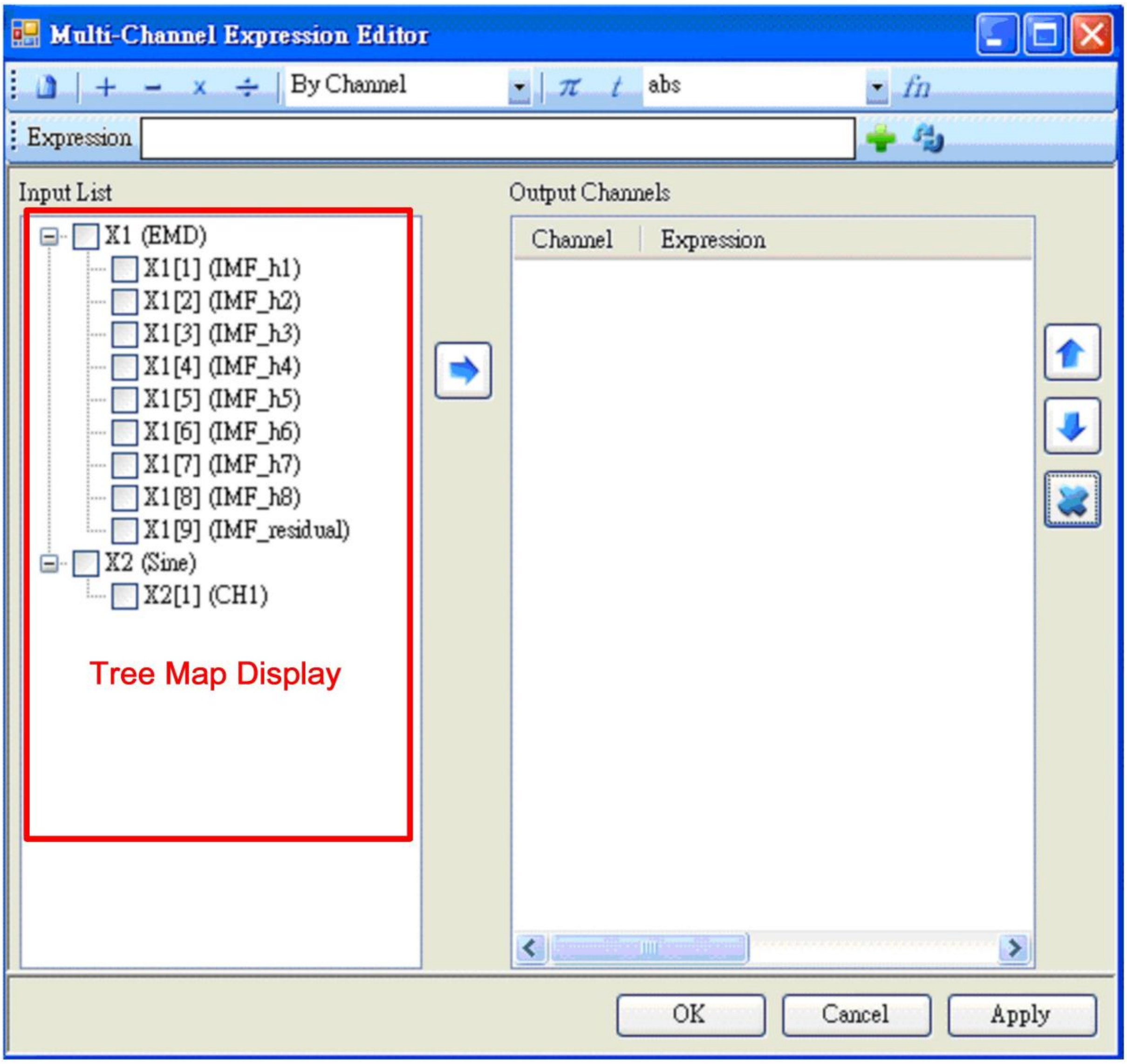




The pop-up window has three panels: Input Signal List, Toolbox, and output Signal List. The operation procedure is as following: select the signal from the Input Signal List, define math equation in the Expression field of Toolbox, the calculated result becomes one of the Output Channel in the Output Signal List. Below explains each of the functionality:

Input List:

Input List displays the signals connect to the input end of the *Math* SFO. By default, the input signal is multi-channel and each channel is displayed as a tree map shown in the graph. The 1st input signal is denoted as X1 in the Expression of Toolbox, the 2nd signal is denoted as X2, etc. The number in the square bracket represents the channel sequence of the input signal. For example, X1[1] represents the 1st channel of the 1st input signal. In addition, X1[1] can be abbreviated as X.



a. Double click a signal in the tree map. It is added to the Expression field.

b. If there is no calculation applys to the select input (output is the same as the input), then press  button to move the signal to the Output List.

c. When checking the checkboxes in the tree map, multi-channels can be selected for complicated calculation. In addition, the signal code (such as X, X1[2]) and math operations (such as +, -, …, sin, log) can be typed directly in Expression field.

Toolbox

The above image shows the Toolbox of MultiChannel Expression Editor. Expression field is used for editing math equation.

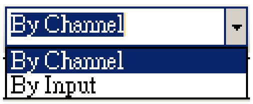
 clear all the output signals in Output Channel panel

 add the math equation in Expression field to the Output Channel list

 replace the math equation in one of the output signal of the Output Channel list with the equation in Expression field

Other buttons in the Toolbox are explained below.

Basic Operation	**Function Definition**
+	Add plus operation to Expression field, “+” can be typed in directly
-	Add minus operation to Expression Field, “-” can be typed in directly
*	Add multiply operation to Expression Field, “*” can be typed in directly
/	Add divide operation to Express Field, “/” can be typed in directly

Special Operation	**Function Definition**
	Set group operation type: By Channel or By Input. By Channel: channel-by-channel calculation for selected multi-channels, the output is a single channel, such as Y[1]=X1[1]+X1[2]+X2[1]+X2[2]. By Input: input-by-input calculation for selected input signals, the output is multi-channel signal, such as Y[1]=X1[1]+X2[1], Y[2]=X1[2]+X2[2].

π	Add π(pi) value to Expression field
t	Add vertor of time axis t to Expression field. It is corresponding to the time axis of the selected input signal.
 	These two tools work as a group. The pull-down menu gives the internal functions. After selecting internal function, press  button to add selected function to Express field. The function can also be typed in directly, such as sin(X1[1]), abs(X1[1]).

Common internal funtions are listed below.

Function	Description	Function	Description
abs	Absolute value	ceiling	Round to the nearest integer toward infinity
floor	Round to the nearest integer towards minus infinity	round	Round to the nearest integer
sin	Sine	asin	Inverse sine
cos	Cosine	acos	Inverse cosine
tan	Tangent	atan	Inverse tangent
sinh	Hyperbolic sine	cosh	Hyperbolic cosine
tanh	Hyperbolic tangent	exp	exp(x) equals e^x
log	Natural logarithm	log10	Base 10 logarithm
pow	pow(x, a) equals x^a	sqrt	Square root
square	Equals x^2	sign	Signum function. Returns 1 if greater than zero, 0 if equals zero and -1 if less than zero.
truncate	Round to the nearest integer toward zero. If x<0, truncate(x) equals ceiling(x). If x>=0, truncate(x) equals floor(x).		
conj	Complex conjugate. For a complex x, conj(x)=Real(x) - i*Im(x).		

In addition, there exists “>”, “<”, “>=”, “<=”, “==”, and “!=” conditional signs. If the condition is satisfied, return 1. If the condition is not satisfied, return 0. There are examples below to show the usage.

Output Channels:

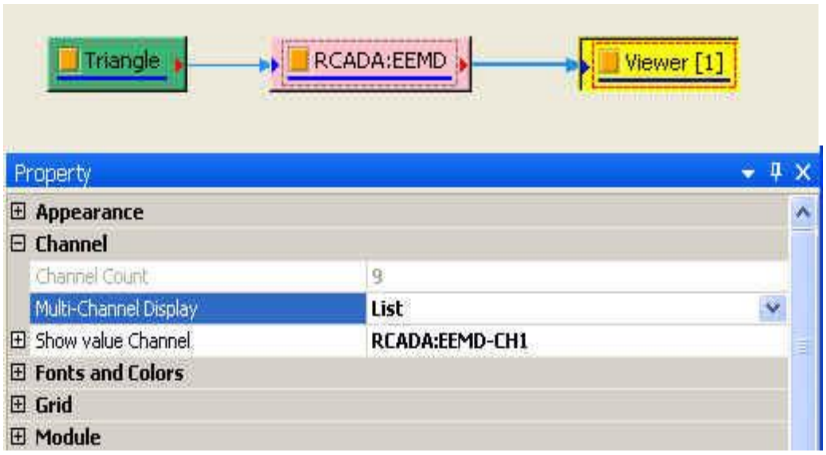
Output Channels display all output channels and defined math equations in Expression. The order of the channels in the output gives the sequence number of the signal. The sequence number/order of the channel can be changed using the button to the right of the panel, moves up and moves down. deletes channels.

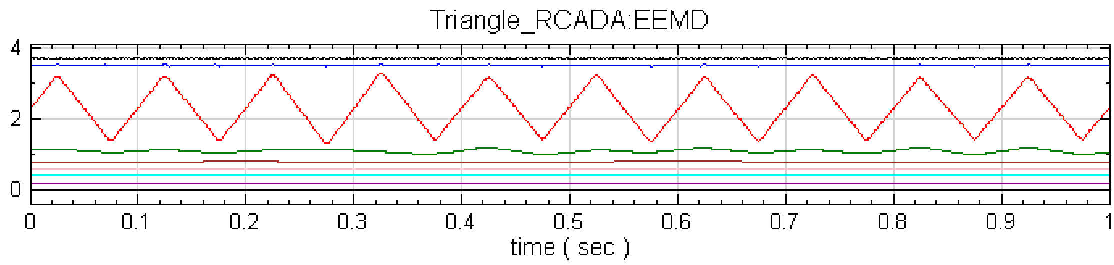
The name of the channel and the equation can be modified. Double clicked the channel to modify the name; if the math equation needs to be modified, select the Express of the target channel, click the mouse left-button once (similar to double click, but with a slower speed), then the Expression can edited directly.

Example

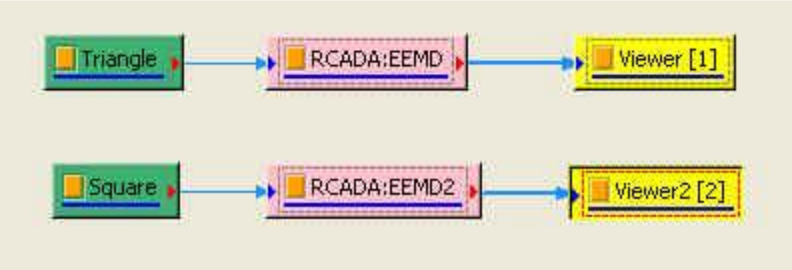
The usage of different calculation method and functionality is shown below.

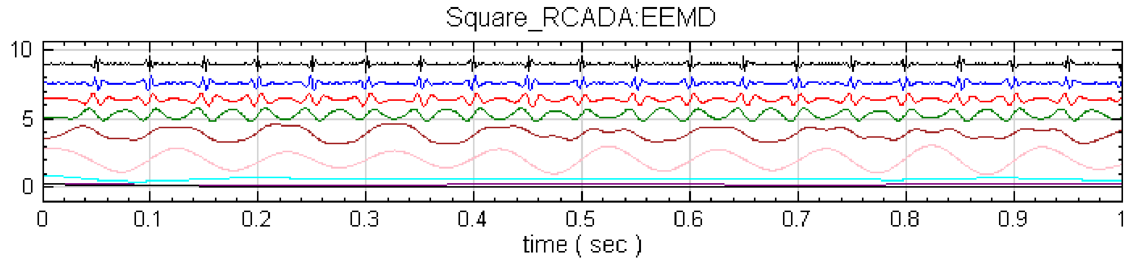
Create a triangle signal using *Source / Triangle Wave,* connect to *Compute / HHT / RCADA : EEMD* for calculation. The goal is to get a Multi-Channel signal. Show the signals using *Viewer / Channel Viewer.* Change *Properties/Multi-Channel Display* in *Channel Viewer* to List. And each channel is displayed separately.



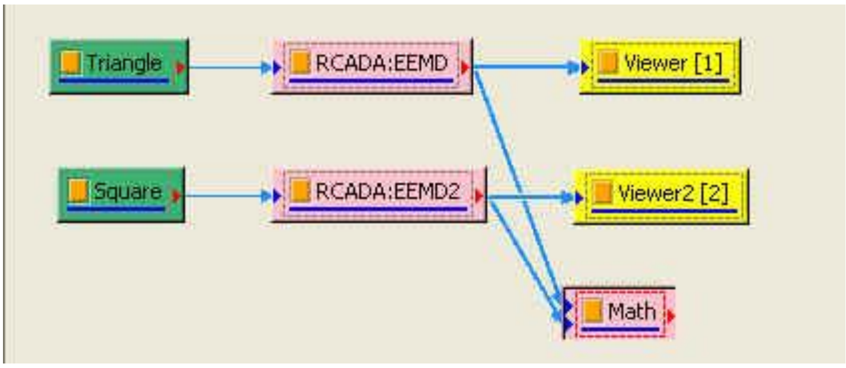


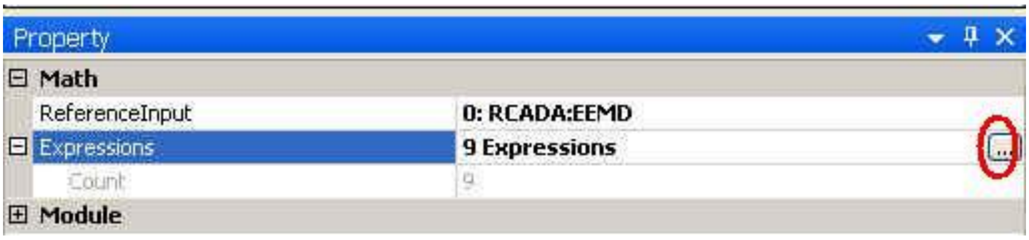
Do the same as step 1, however, change Triangle Wave to Square Wave.



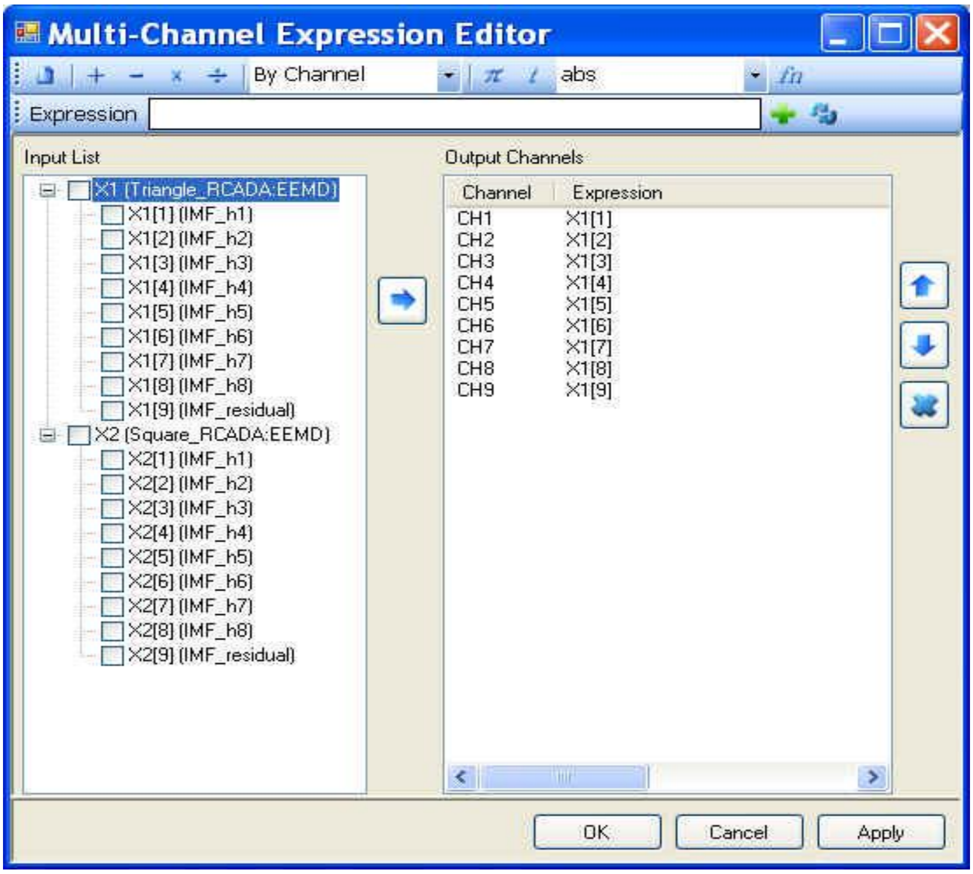


Connect RCADA : EEMD and RCADA : EEMD2 to *Compute / Mathematics / Math,* expand + box before Expressions in Math Property Window and click Properties / Expression Editor at the end of the line to open the interface.

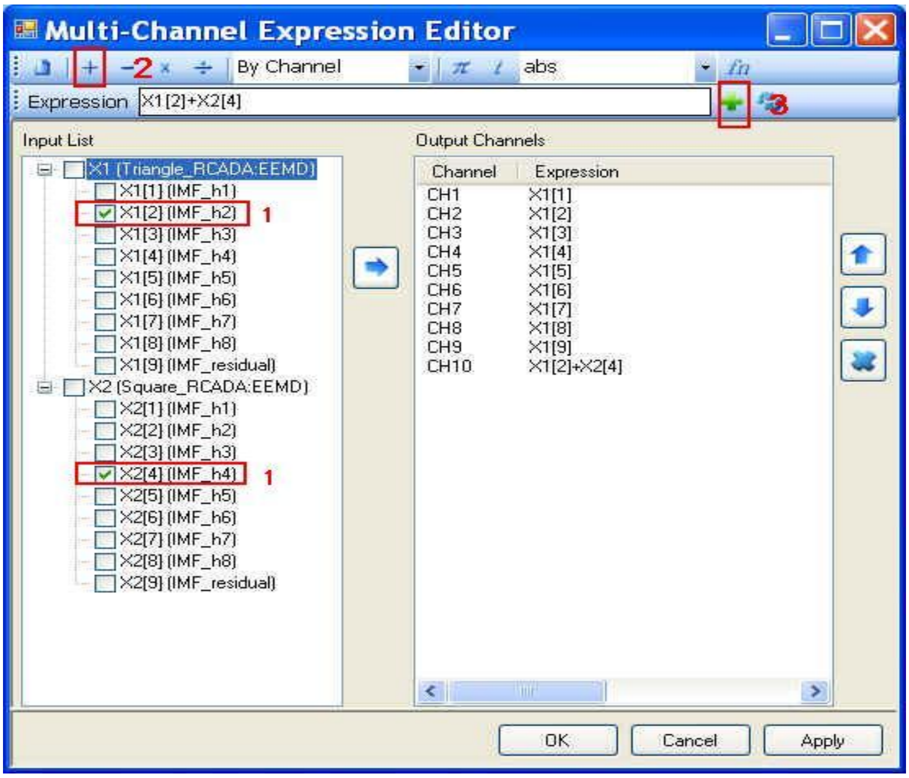




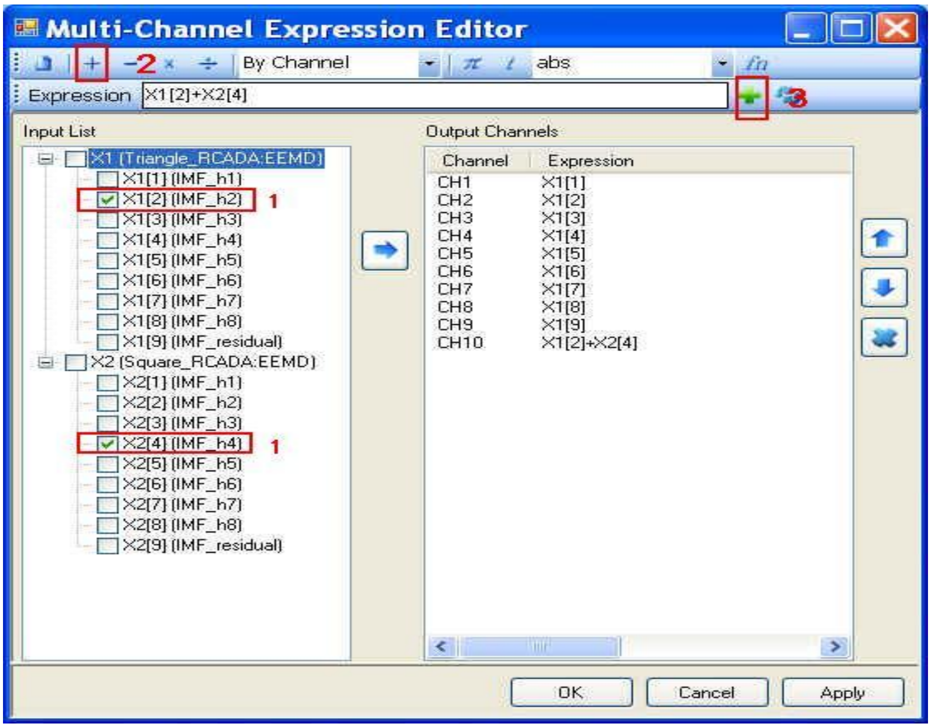
In Multi-Channel Expression Editor, open the tree map in the Input List by clicking + sign in front of the signals. Please note the default output signal is the 1st input signal.



In order to sum channel 2 of X1 and channel 4 of X2 together, select both channels and click basic operator “+”. The summation equation is added to the Expression field. This equation “X1[2]+X2[4]” can be typed in the field directly.

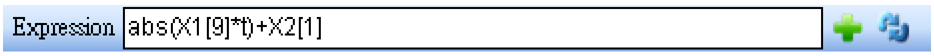


Press button to transfer the equation in Expression field to Output Channels. Here CH10 is added.

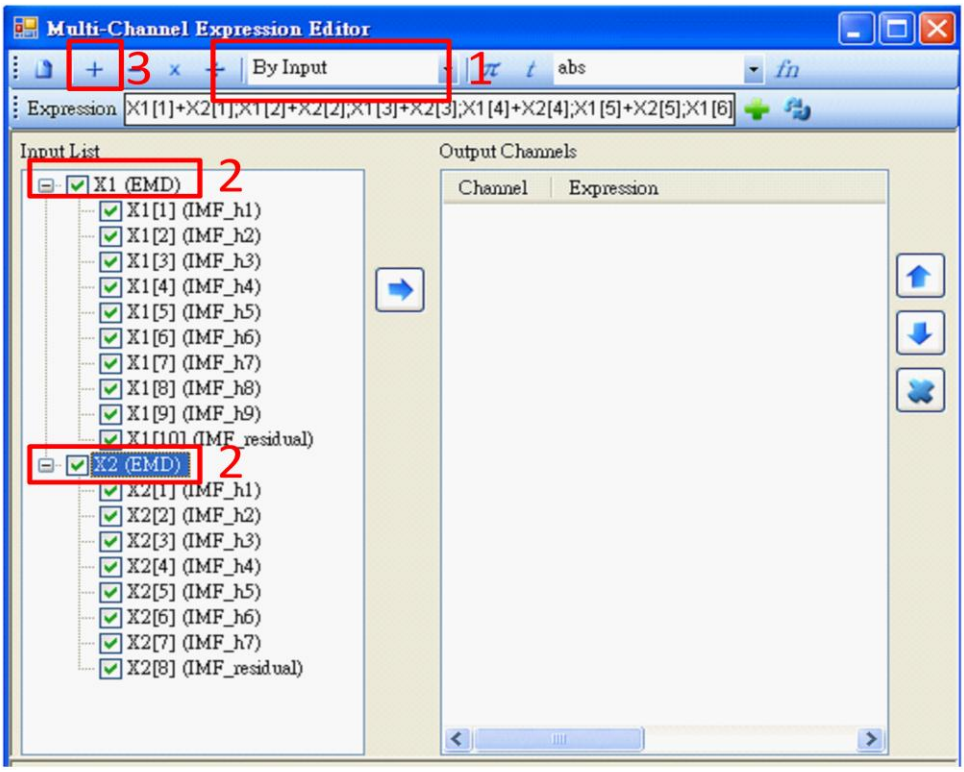


Next, the channel 9 of X1 multiplies corresponding time t, then add channel 1 of X2. Add channel 9 to Expression field by double clicking X1[9] under X1, then click basic operator in the Toolbox to complete the equation. Surely, the equation can be directly typed in the Express field "X1[9]* t + X2[1]".

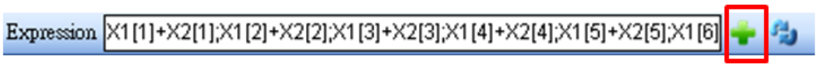
To get the absolute value of X1[9]* t, edit the equation directly to "abs(X1[9]* t)+ X2[1]", or highlight "X1[9]* t" part in Expression field, then select function abs from the function list and press *fn* button to complete the equation. All internal functions can be added this way. Finally, click button to transfer the equation to the Output List.



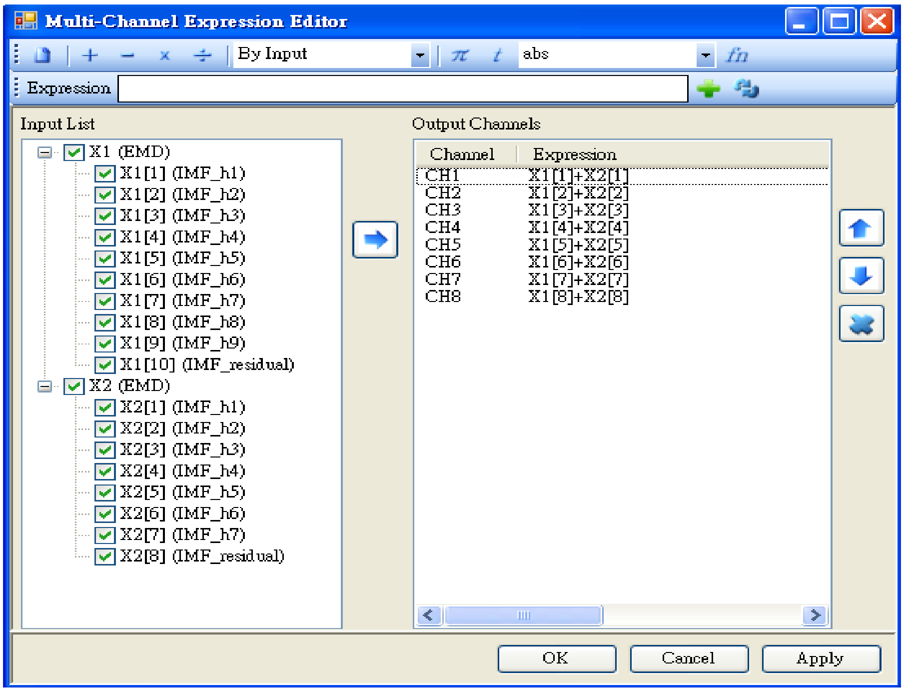
For the calculation of more than two input signals, such as CH1+CH1, CH2+CH2, "By Input" option can be deployed. So the top level selection is input signals instead of each channel under the signal. Once the signal is selected, all channels under it are selected automatically. As shown below, select input signal X1 and X2, then select "By Input", and click basic operator "+", the full calculation equations are added to the Expression field.



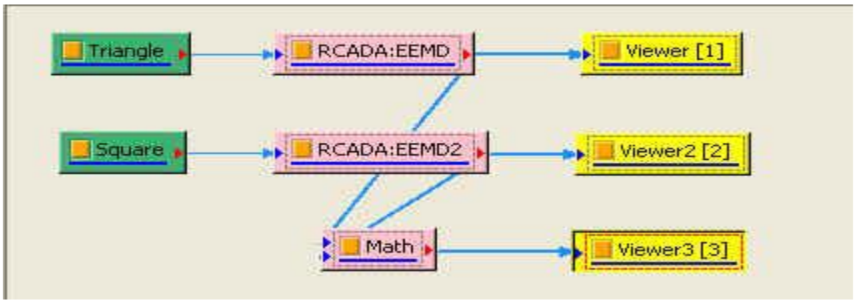
Finally click button to transfer the equations to the Output Channels.

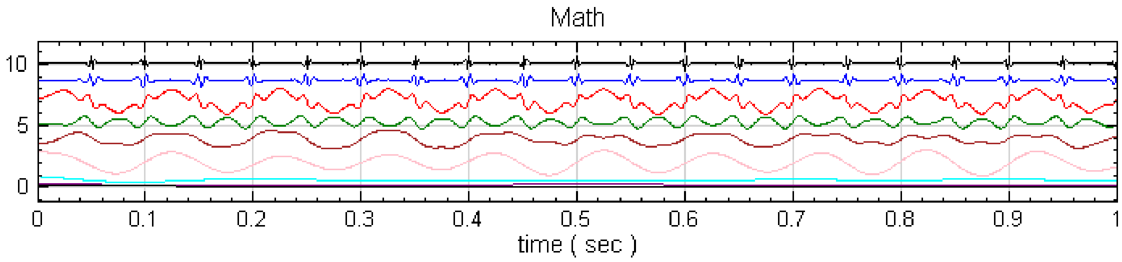


In Output Channels panel, there are 8 calculation channels added. Since there are 10 channels in X1 and 8 channels in X2, Math uses the less number for the output, i.e. CH1 ~ CH8.

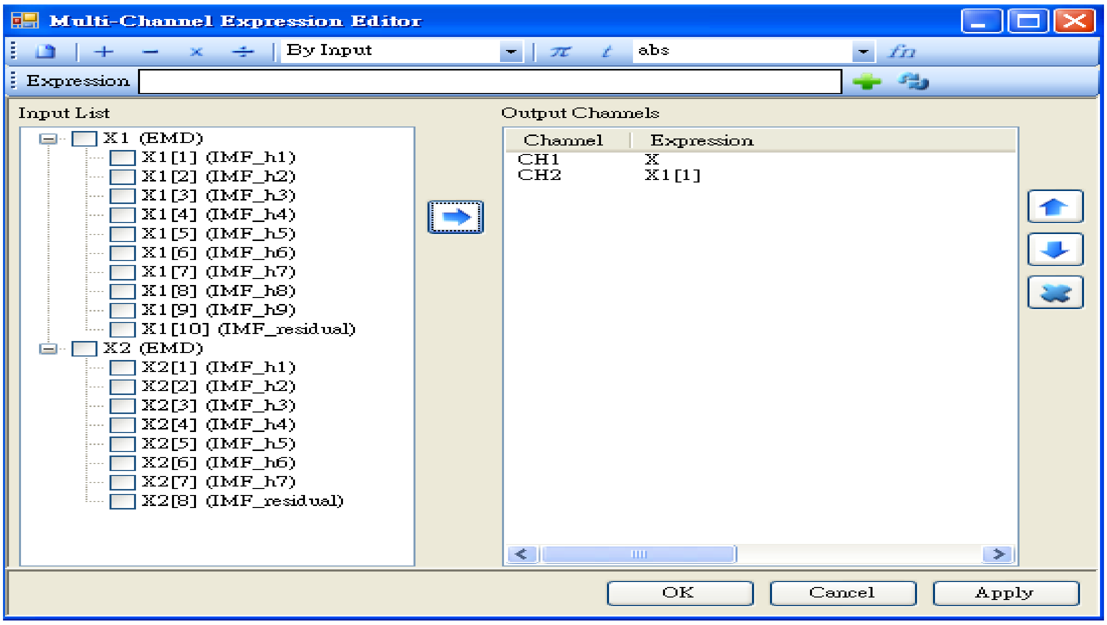


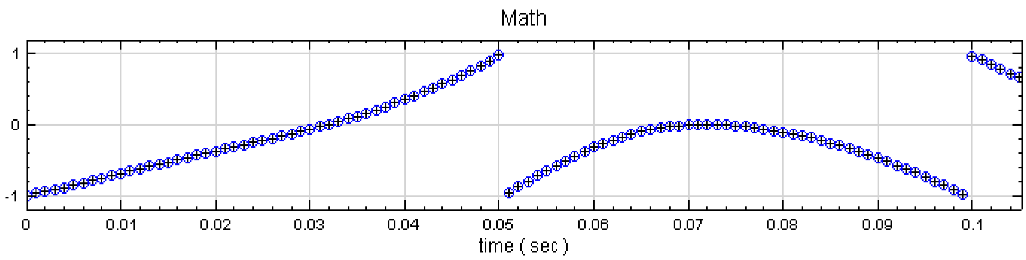
To the right of the Output Channels, there are 3 buttons. They can be used to move the output channel up and dow, or delete output channels. Once the output channels are ready, press “OK” or “Apply” to complete. The Output Channels can be displayed via *Channel Viewer.*



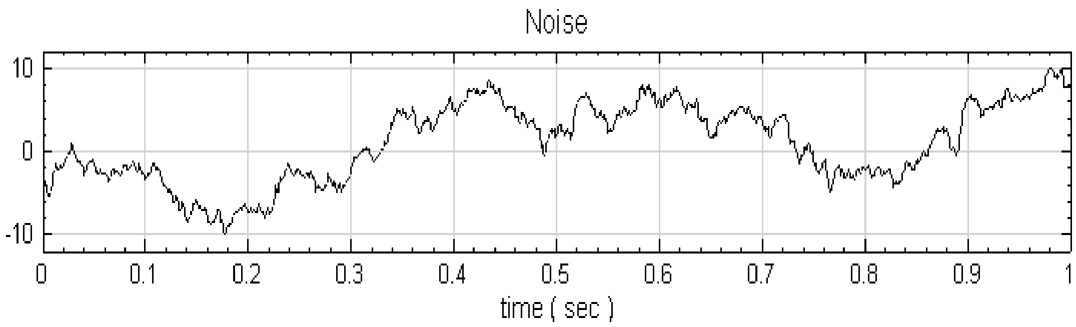


Set output signals to be X and X1, press “Appy” and view the result with *Channel viewer*. Both signals are the same since X = X1[1].

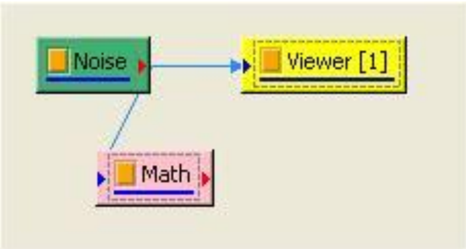




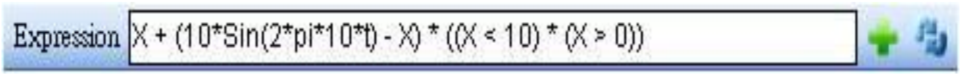
Also the Expression can carry out calculation for ">", "<", "==", "!=" etc. Create a Noise signal and set NoiseType to Brown, display the signal using *viewer / Channel Viewer.*



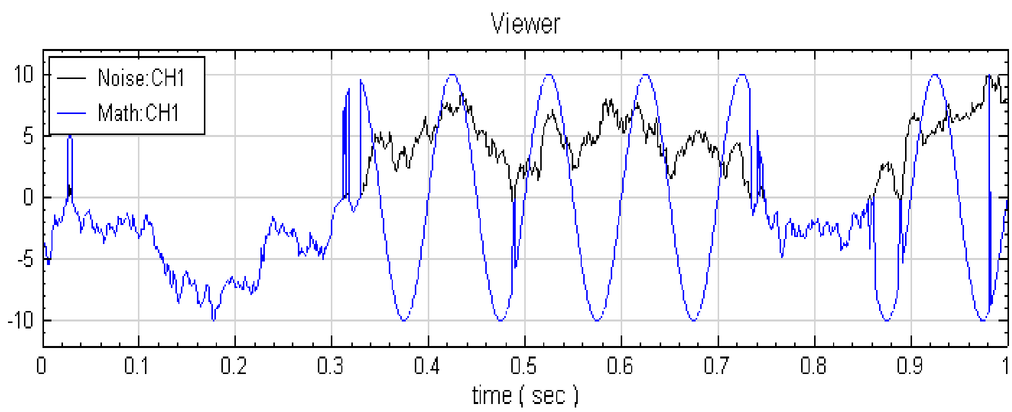
Connect Noise SFO to *Compute / Mathematics / Math*, and display the output signal of Math with *Channel Viewer.*



Replace the Noise signal, where the amplitude is between 0 and 10, with a Sine wave. The equation of the calculation is written in the Expression field,



Display the output of Math in the same viewer as the input Noise signal and compare the curves.



Related Functions

Viewer, Mixer, Multiplier, Source.

3.3.5 Diff

This component performs subtraction or differentiation operation on two signals.

Introduction

Let $X = \{x_0, x_1 ... x_{N-1}\}$ be a length-N signal, the various difference/differentiation is defined as below.

- Forward difference：

$$\Delta x_i = x_{i+1} - x_i$$

Divided by the sampling period h, the approximate differentiation value is given by

$$\frac{\Delta x_i}{h} = \frac{x_{i+1} - x_i}{h} \cong \frac{dx_i}{dt}$$

- Central difference：

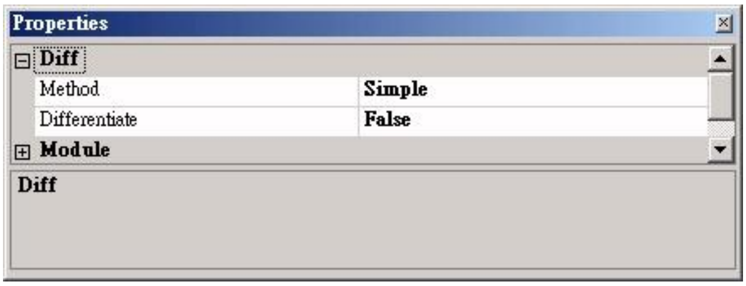
$$\Delta x_i = \frac{x_{i+1} - x_{i-1}}{2}$$

Dividing the central difference by sampling period, the approximation of differentiation can be obtained as follows.

$$\frac{\Delta x_i}{h} = \frac{x_{i+1} - x_{i-1}}{2} \frac{1}{h} \cong \frac{dx_i}{dt}$$

Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular) and Audio (which could be real number or complex number, single channel or multi-channel, Regular). Settings of related properties are given below.

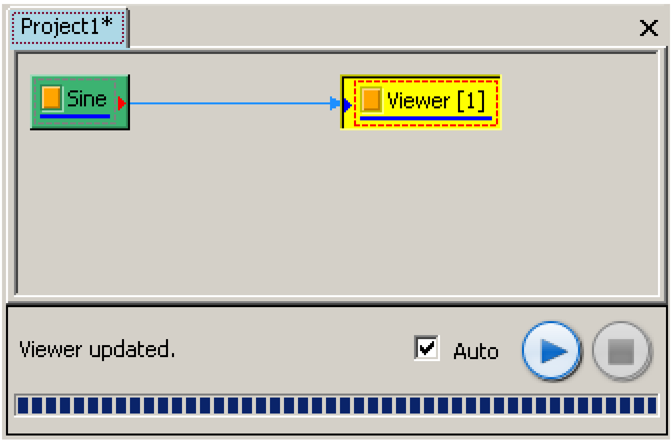


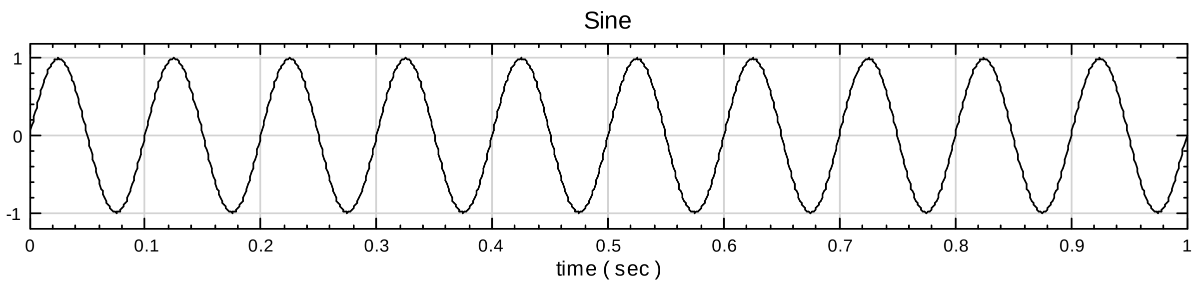
Property Name	Property Definition	Default Value
Method	The differentiation methods include Simple and Symmetrized. Simple is 2 points forward difference while Symmetrized is central difference.	Simple
Differentiate	Divide the result by the sampling period to obtain the approximation of differentiation	False

Example

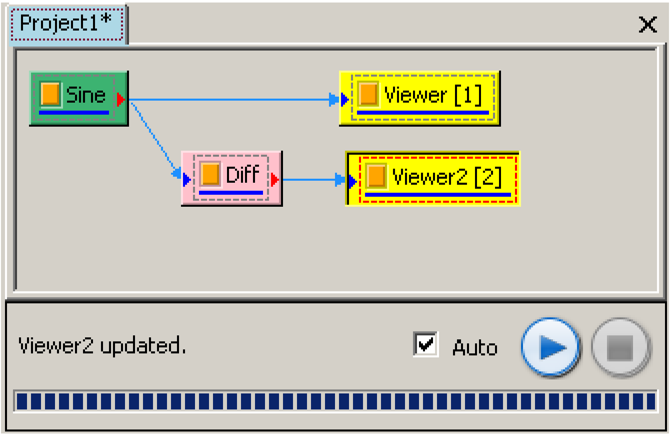
This example shows the differentiation of a Sine Wave.

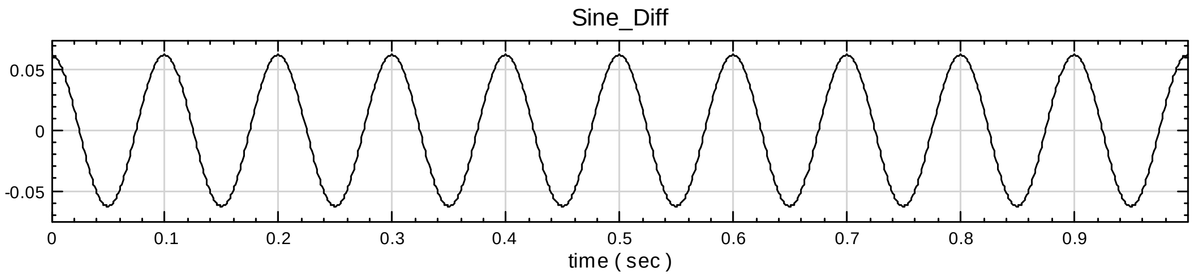
1. Right click in Network Window to select *Source→Sine Wave* to generate a sine wave, and then use *Viewer→Channel Viewer* to show it in the window.



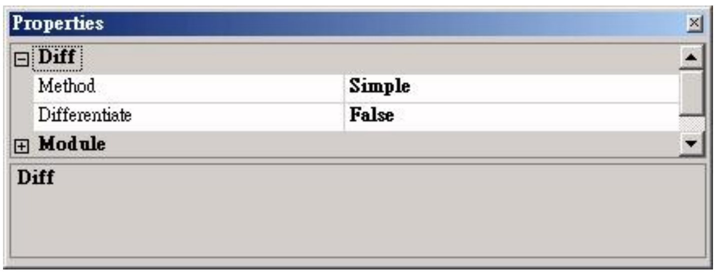


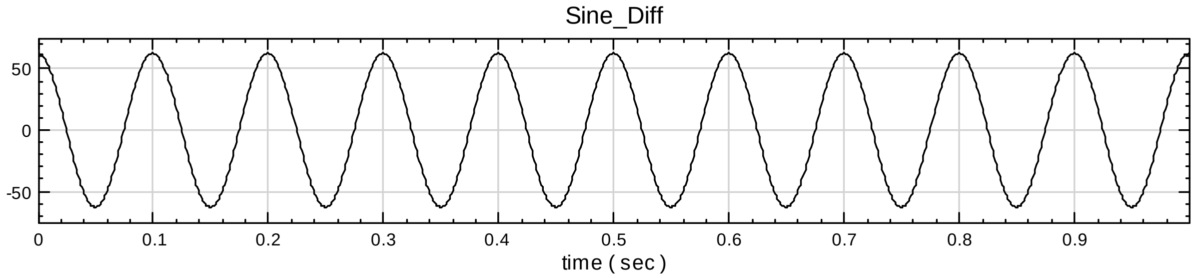
2. Right click on the icon of *Sine*, select *Computer→Mathematics→Diff*, and then use *View→Channel Viewer* to plot the calculation result, as shown below. It can be seen that the Sine wave is changed to Cosine after Diff calculation. However, because the default value of Differentiate is False, the amplitude is very small.





3. The approximation of differentiation can be obtained by changing the *Properties/Differentiate* in Diff to True. Here, the *Source→Sine Wave* is $\sin(2\pi f t)$, where *f* is the signal frequency, *t* is the signal time, and the differentiation of this sine wave should be $2\pi f \cos(2\pi f t)$. In this example, f is 10 Hz, 2π is 6.28, therefore, the maximum amplitude should be $2\pi f = 62.83$. The result can be verified by comparing with the result shown below.





Related Functions

Integrate, Viewer.

3.3.6 Integrate

This component performs integration on input signals.

Introduction

Let $X = \{x_0, x_1 ... x_{N-1}\}$ be an N-length signal, $T = \{t_0, t_1 ... t_{N-1}\}$ be the corresponding X-axis (time axis) coordinates, the numerical integration using Simple can be denoted as the formula below.

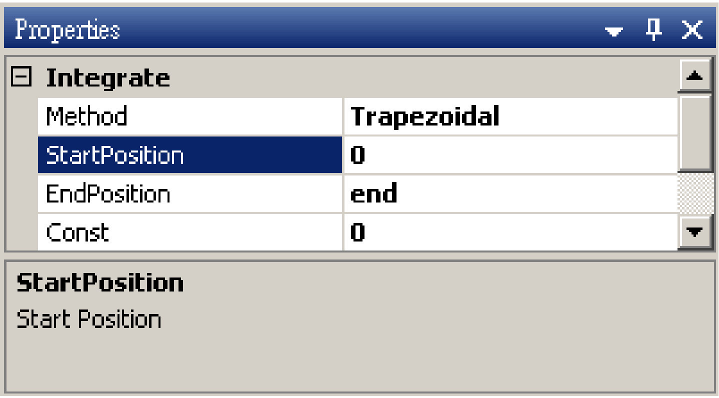
$$y_i = \int_0^{t_i} x_i dt \cong \sum_{k=0}^{i} x_k (t_{k+1} - t_k)$$

If Trapezoidal is used, the formula is denoted as below

$$y_i = \int_0^{t_i} x_i dt \cong \sum_{k=0}^{i} \frac{1}{2} (x_{k+1} + x_k)(t_{k+1} - t_k)$$

Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular) and Audio (which could be real number or complex number, single channel or multi-channel, Regular). The related properties are introduced as in the table below.



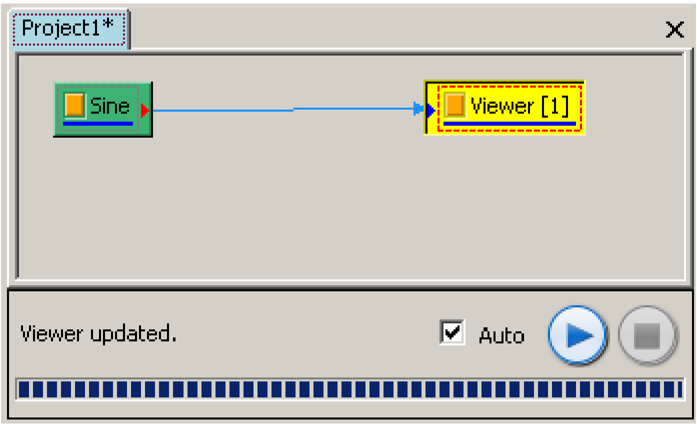
Property Name	Property Definition	Default value
Method	The methods of numerical integration, including Simple and Trapezoidal	Trapezoidal

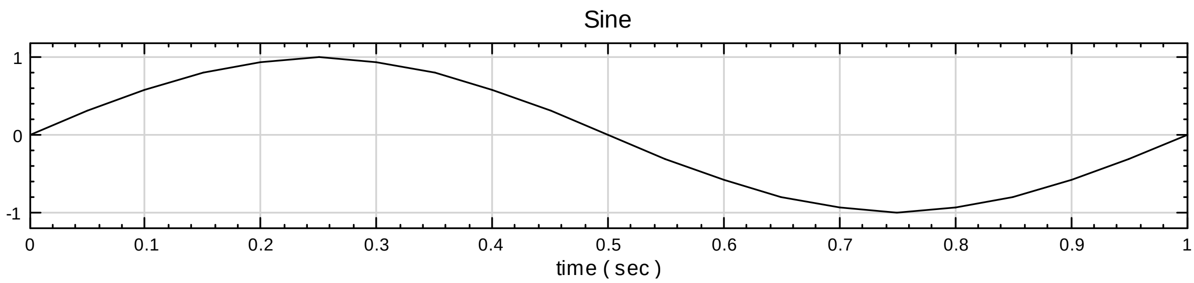
StartPosition	The start position in X-axis for integration	The starting point of the input signal
EndPosition	The end position in X-axis for integration	The ending point of the input signal
Const	The shift along Y-axis after integration	0

Example

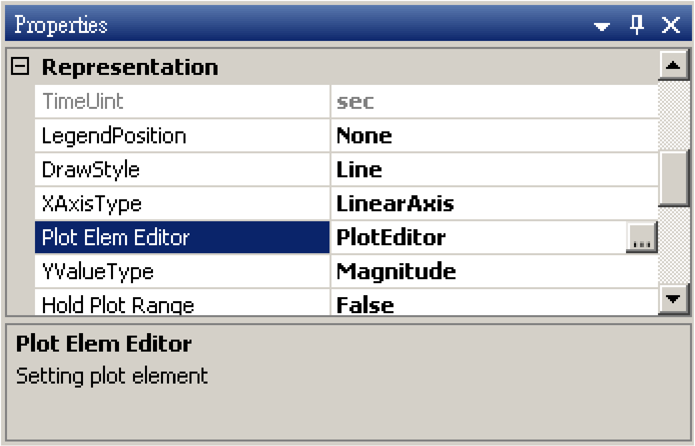
This example shows the integration on a sine wave.

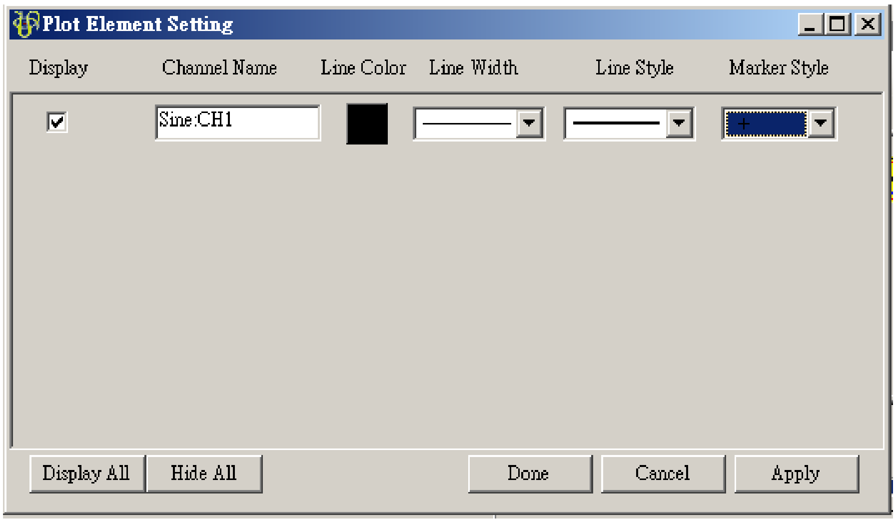
1. Click right button in Network Window, select *Source→Sine Wave* to create a sine wave, change the *Properties/SignalFreq* to 1Hz, *SampleFreq* to 20Hz, *TimeLength* to 1 second (must be set in the final step), and then use *Viewer→Channel Viewer* to show it in the window, as shown below.

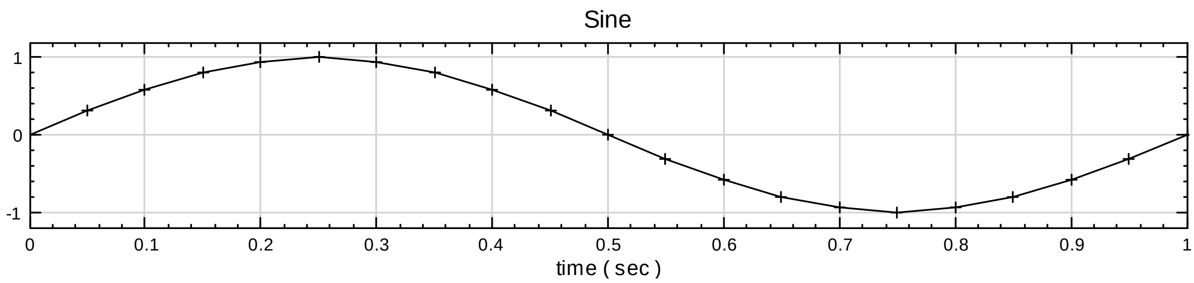




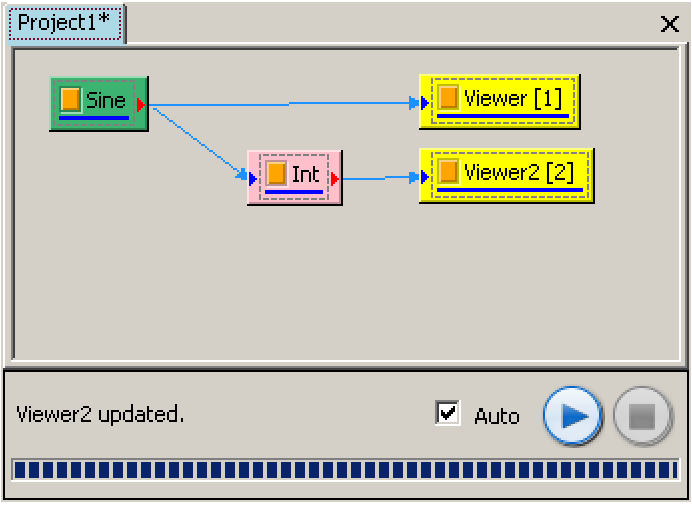
2. To show every points clearly, click *PlotEditor* in the *Properties/Representation→ Plot Elem Editor* which in turn is in the *Viewer[1]*. In the popped-up *Plot Element Setting* window, select 「+」 in *Marker Style* to mark every time points as symbol of 「+」 in the curve.

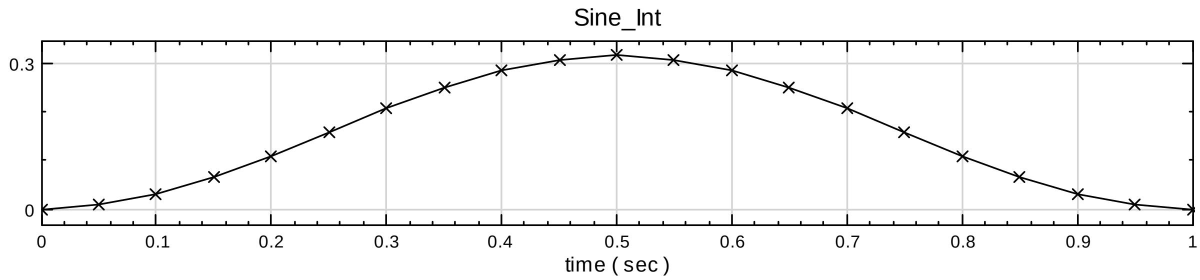




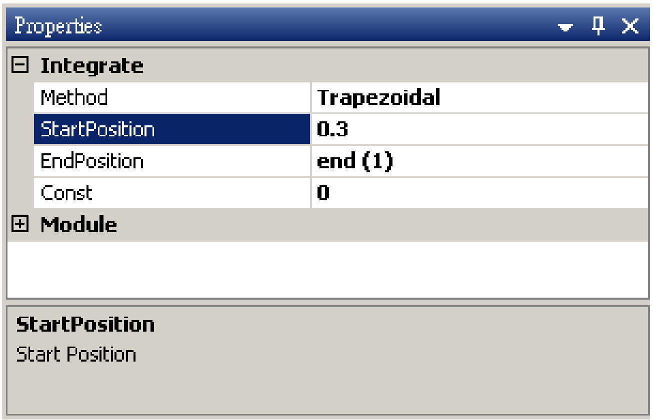


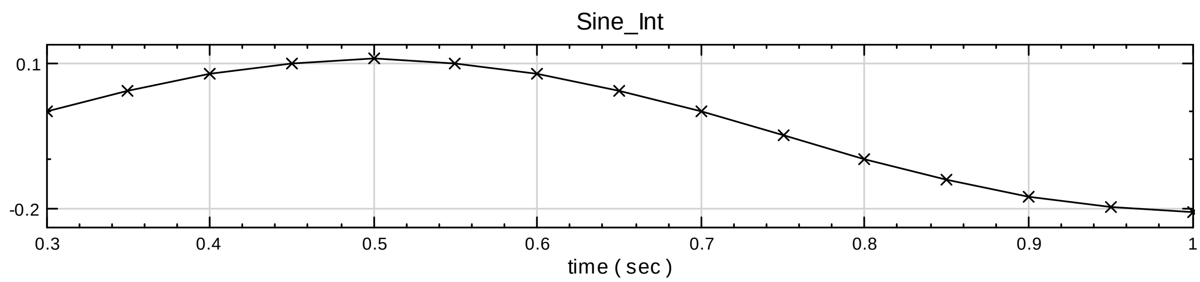
3. Perform numerical integration (Compute→Mathematics→Integrate) on the sine wave, and change *Marker Style* to 「x」 as what has been done in step 1 and 2. The figure plotted is the integration result.

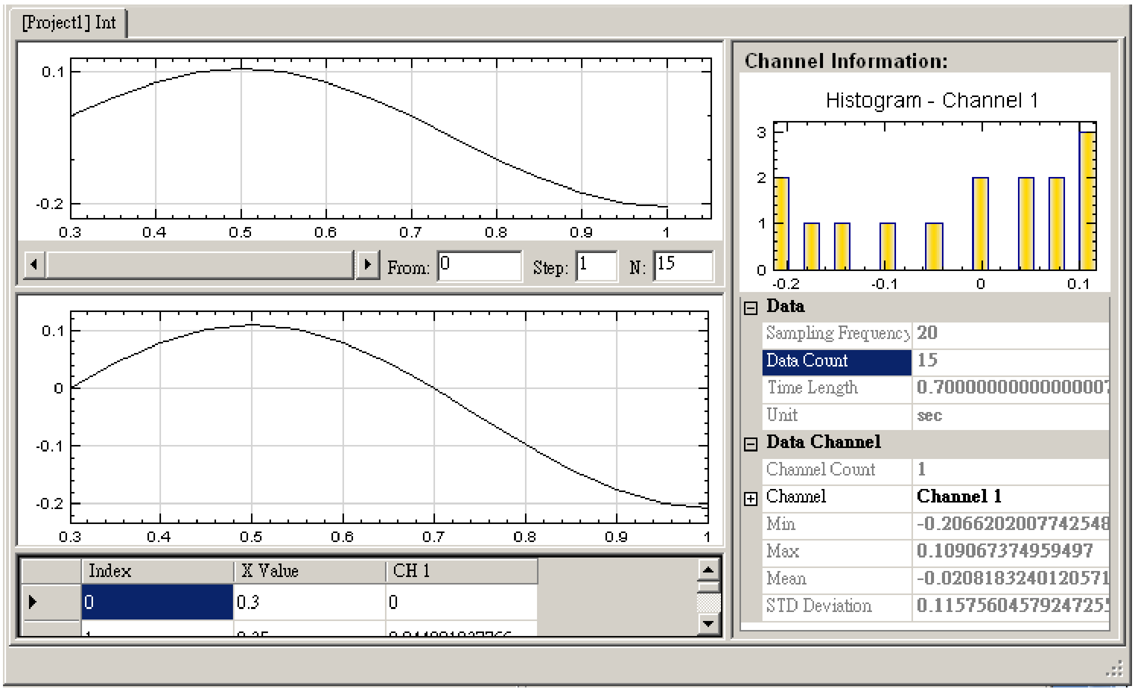




4. Change *Properties/StartPosition* of the Int SFO to 0.3, the new calculation result is shown below. Next, use *DataViewer* to observe the signal output from Int SFO. It can be seen that the original value of 21 in DataCount has been changed to 15. Note: the changing of StartPosition and EndPosition would affect the output signal length.







Related Functions

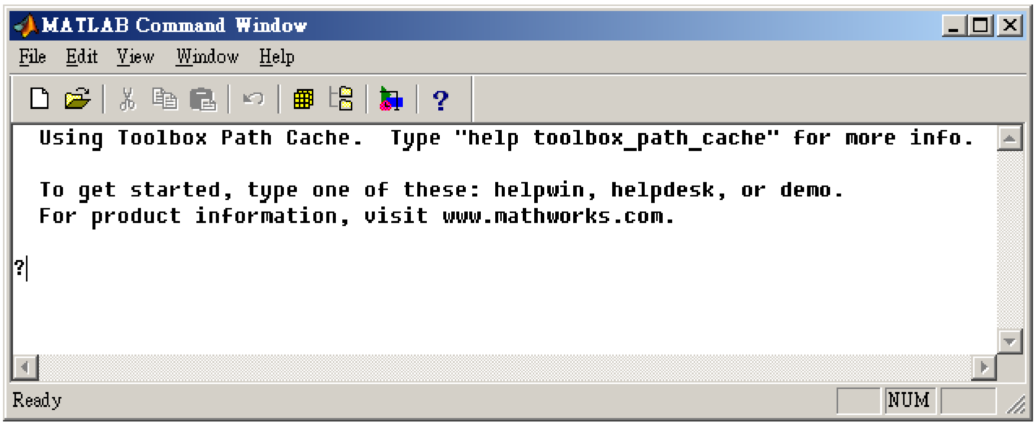
Diff, Source, Channel Viewer.

3.3.7 DoMatlab

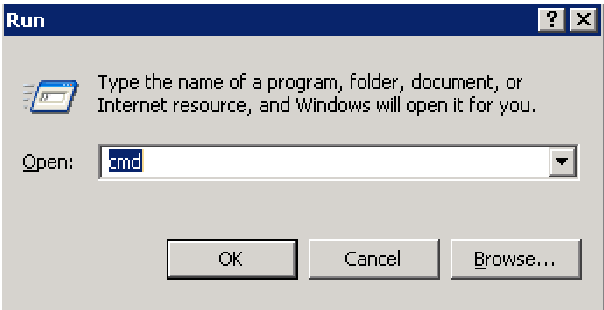
DoMatlab could be used to input the outcome signal data from DataDemon to Matlab engine and run Matlab codes. The Matlab calculation results also could be sent back to DATADEMON.

Properties and Instruction

DoMatlab can accept all output format available in DATADEMON. When this component (DATADEMON) is connected to DoMatlab, the code would run the Matlab Command window automatically. Basic MATLAB functionalities such as Workspace browser, Path browser are provided in Command window. Users may reference to MATLAB introduction files for usage instructions.



Note that MATLAB, which is a product by MathWorks, must be installed in the system before DoMatlab could be run. If DoMatlab fails to run even after Matlab installation, the reason most likely is that the MATLAB is not registered as COM server. To solve this issue, on WINDOWS systems, type *cmd* in [START]→[RUN] window to startup the *command line* console.



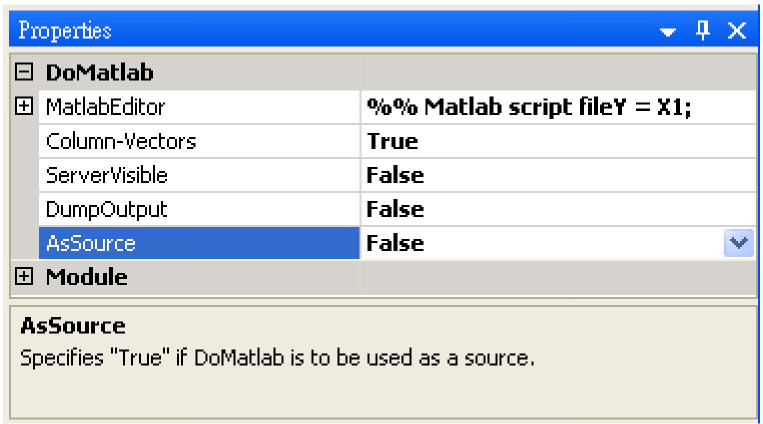
Next, type [matlab /regserver] in the command line and hit enter. The issue should get fixed now.

If the issue persists, please try the following two steps:

- Right click 'Properties' on 'My Computer', System Properties window pops up. Choose 'Environment Variables' at the bottom of Advanced tab. Then click New button for User Variables and add New User Variable,
  Variable name: MATLAB_RESERVE_LO
  Variable value: 0

- Select 'Path' under System variables panel, then click Edit; and add string ';C:\Program Files\MATLAB\R2010a\bin\win32' to Variable value entry. The string is the installation path for Matlab, please change it accordingly.

Properties in DoMatlab are shown in the figure below and the corresponding definitions are listed in the table below.

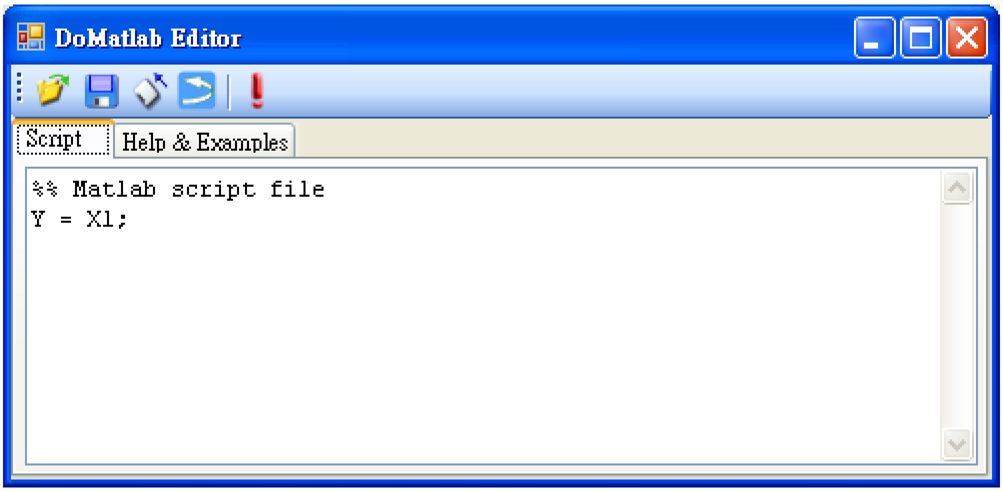


Property Name	Property Definition	Default Value
MatlabEditor	This window is similar to the M file editor. At the right side, there is a button to run the DoMatlab Editor which is used to write MATLAB code or	None

	call matlab functions.	
Column-Vectors	Use to determine whether the data is Column-wise when signals are passed to MATLAB	True
ServerVisible	Use to show and hide MATLAB Command Window	True
DumpOutput	Use to show and hide the information generated during Matlab processing	False
Buffer Length	When DumpOutput is True, this property is provided to set the maximum number of data that can be sent back from Matlab	5000
ViewBuffer	When DumpOutput is True, this property is provided to pop up a window to show the Matlab messages generated during the processing	None
ReferenceInput	When there are multiple input signals, this property decides which input signal is used as reference signal for the default time axis (Y_DESC) of output variable	The 1st input signal
AsSource	Determine whether DoMatlab should be changed to a signal source which can be used to generate signal data	False

Matlab Editor is the code editor for DoMatlab. The details of basic setting for DoMatlab, e.g., input and output, are given below.

Introduction of Matlab Editor Interfaces



DoMatlabe Editor Appearance is shown as above and the functionalities of buttons are introduced below.

1. Button represents opening M file.

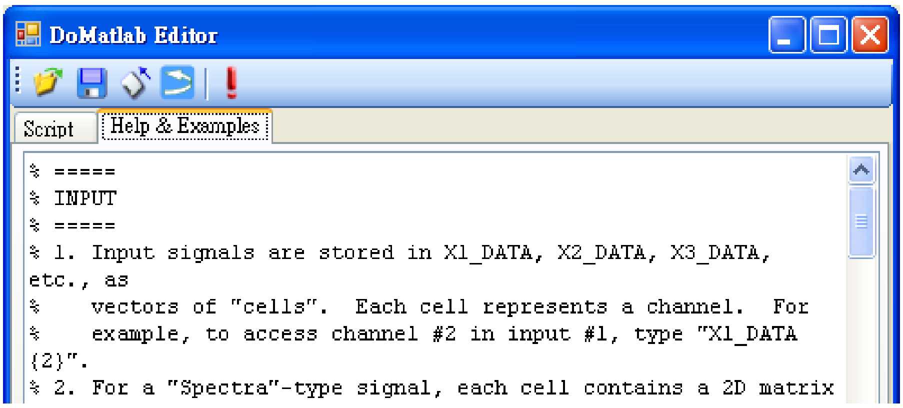
2. Save the content in DoMatlab Editor in a M-file

3. Turn off the editor

4. Clear contents in Script area

5. Is the button to run the script

In addition, there are two tabs in DoMatlab Editor: Script and Help & Example. Script is the workspace for users to write MATLAB codes. The default instruction is "Y=X1". The Help & Example page detailes the input and output format in DoMatlab and provides simple examples.



The variable format of input signals

DoMatlab allows multiple input sources which are defined as X1, X2…Xn following their input order. When a new signal adds to the input, DoMatlab would add 5 variables which are X, Xn, Xn_DATA, Xn_DESC, Xn_Freq. DoMatlab only allows one output signal which is defined as variable Y. The meanings of variables mentioned above are explained below.

X: It is a default variable used to facilitate user's operation. It is defined as the value of the 1^{st} channel in the first variable, i.e. X=X1_DATA{1}.

Xn_DATA: It is defined as the complete input signal value of the n^{th} input signal. Because a signal may have multiple channels, the values are saved using cell array. Different channel data are stored in different cells following their order.

Xn: It is defined as the signal value of the 1^{st} channel of the n^{th} input signal, i.e., Xn=Xn_DATA{1}. This data is saved in a double array.

The input signals are divided into 3 types, which are Signal (time series and spectrum analysis result), Spectra (time-frequency analysis result) and numeric data (Numeric, e.g., the calculation result of Basic statistics module). Methods of saving these 3-type signals in variable of Xn_DATA are shown in the table below. Note that the property of Column-wise could change the definitions of row and column of signals in DoMatlab.

Data type	Format
Signal (Signal, Spectrum)	Column-wise is True: *m x n* double array or complex double array Column-wise is False: *n x m* double array or complex double array Where *m* is the data length and *n* is the number of channels
Time-frequency analysis (Spectra)	Column-wise is True: *m x n* complex double array, the columns represent different time, the rows represent different frequency, and every element represents the time and frequency of the corresponding position Column-wise is False: *n x m* complex double array, the columns represent different frequency, the rows

	represent different time, and every element represents the time and frequency of the corresponding position Where *m* is the length in the discrete time space, *n* is the length in the discrete frequency space. Note that the spectra data only support single channel

Data type	Format
Numerics	Column-wise is True: *m x n* double array or complex double array Column-wise is False: *n x m* double array or complex double array Where *m* is the number of rows and *n* is the number of columns. Both are for original input data.

The definition of structure fields in Xn_DESC are shown below.

Field Name	Definition	Format
name	Name of the input signal	Char [1·length]
type	Signal, Numeric, Spectra. Type of signal.	Char [1·length]
channel count	The number of input signal channel	Integer
channel names	Name of channel, e.g. CH1	Char array [nch·max(length)]
lengths	Data length of a channel	Integer array [ndim·1]
starts	The time starting point of signal. It is meaningless for Numeric.	Double array [ndim·1]
intervals	The signal sampling period. It is meaningless for Numeric.	Double array [ndim·1]
units	The unit of signal X-axis(could	Char array

	be unit of time or frequency)	[ndim·max(length)]
formats	The signal time-axis format. Currently, Regular and Indexed are available.	Char array [ndim·max(length)]
coords	X coordinates of input signal. For signal and spectra, it is time. For spectrum, it is frequency.	Double array [ndim·max(length)]

Note: Char is string, Length is the length of byte or number, ndim is the array dimension (if the input is numeric), nch is the number of channels, max is the maximum value.

Xn_FREQ is used to store the sampling frequency of input signal, that is

$$\text{Xn\_Freq} = \frac{1}{\text{Xn\_DESC.intervals}(1)}$$

If the data is related to sampling, such as Signal and Spectra, the sampling frequency would be stored in Xn_FREQ. If the data is unrelated to sampling, such as Spectrum and Numeric, number 1 would be stored in Xn_FREQ.

Storage format of output signal variable

Variable Y is the output variable of DoMatlab, using the storage format defined in Y_DESC whose content is the same as that saved in the Xn_DESC. The default value of Y_DESC is identical to the input signal format defined in *Properties/ReferenceInput* of DoMatlab. The users can also change Y_DESC manually. DoMatlab determineds the type of output signal based on Y_DESC. The table below shows the meaning of each field.

Filed Name	Introduction	Necessity	Default Value
name	The name of the input signal	Optional	DoMatlab
type	The type of signals. Currently, available types are Signal, Numeric, Spectrum, Spectra.	Optional	Signal
channel names	Channel name, e.g. CH1.	Optional	None
starts	The starting point of signal. It is	Optional	0

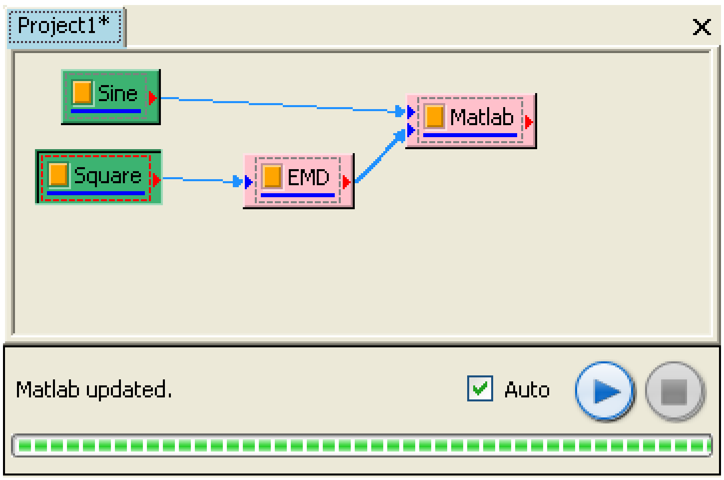
	meaningless for Numeric.		
intervals	The sampling period. It is meaningless for Numeric.	**Required**	None
units	The unit of signal (time or frequency)	Optional	Sec or Hz
formats	The time axis format of signal. Currently, Regular and Indexed are available.	Optional	Regular
coords	The X-axis value of input signal. For signal and spectra, it is time. For spectrum, it is frequency.	**Required** (if the Format is Indexed)	Indexed

Example

The three examples below show the DoMatlab operation.

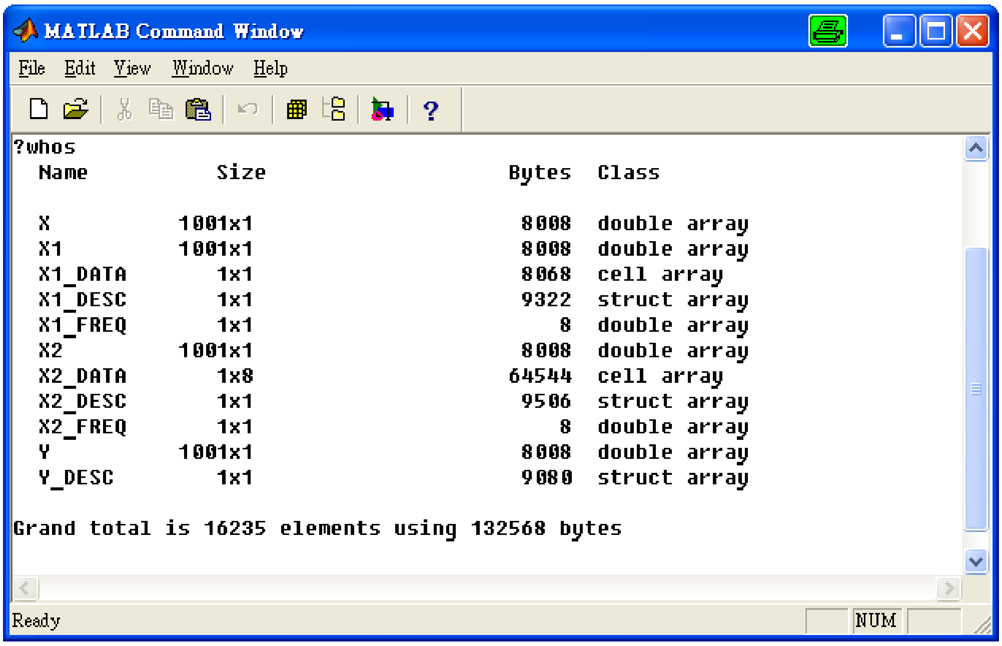
The basic variable structure in DoMatlab

1. Use *Source→Sine Wave* and *Square Wave* to generate two groups of signals. Connect the square wave to Compute→HHT→EEMD to create a multi-channel signal. Then connect both of Sine and EEMD to DoMatlab.

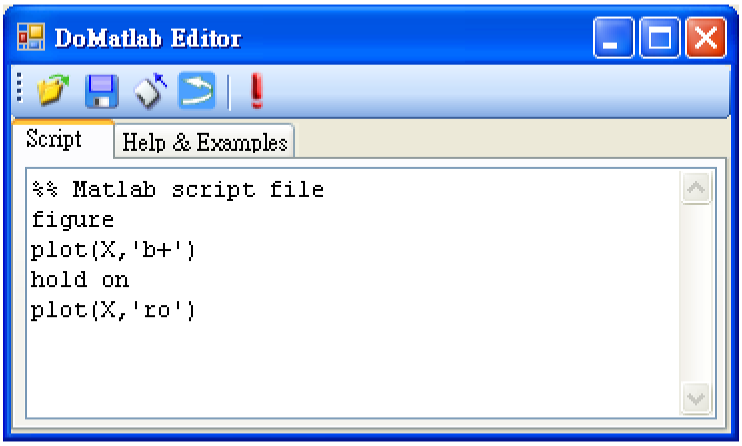


2. Automatically, a Matlab command window pops up after DoMatlab component is generated. Firstly, enter “whos” in the Matlab command window, or press the

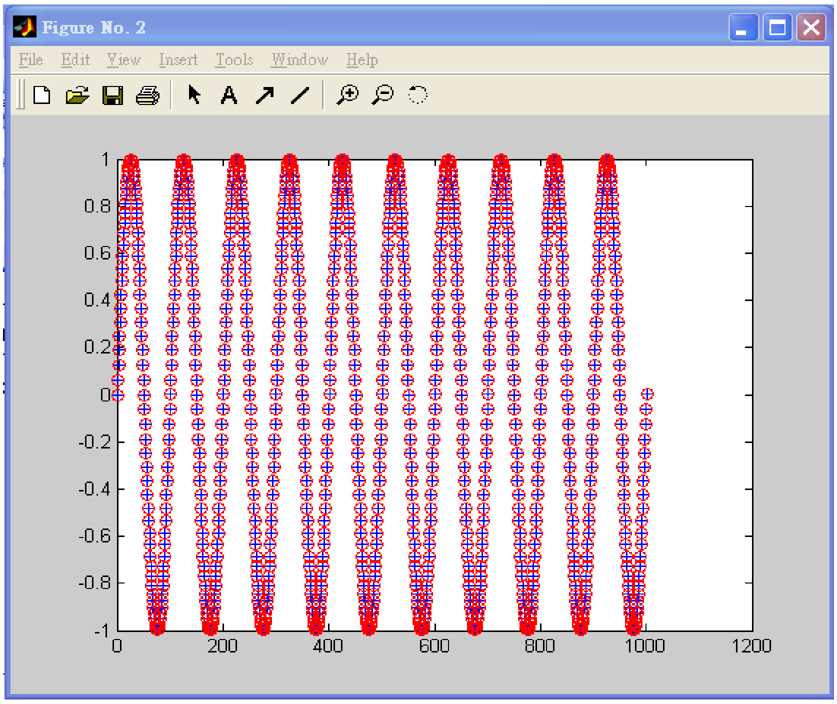
Workspace browser button in the window tools to check the current Matlab variables available.



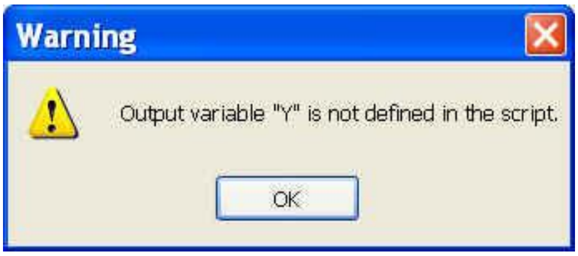
Because there are two set of input signals, DoMatlab defines two variable groups of X1 and X2 where X1 is the sine wave of the signal channel and X2 is multi-channel output signal created by EEMD. The default variable X is the 1st channel of the 1st input signal. We can use the “plot” in Matlab script editor to plot X and X1.



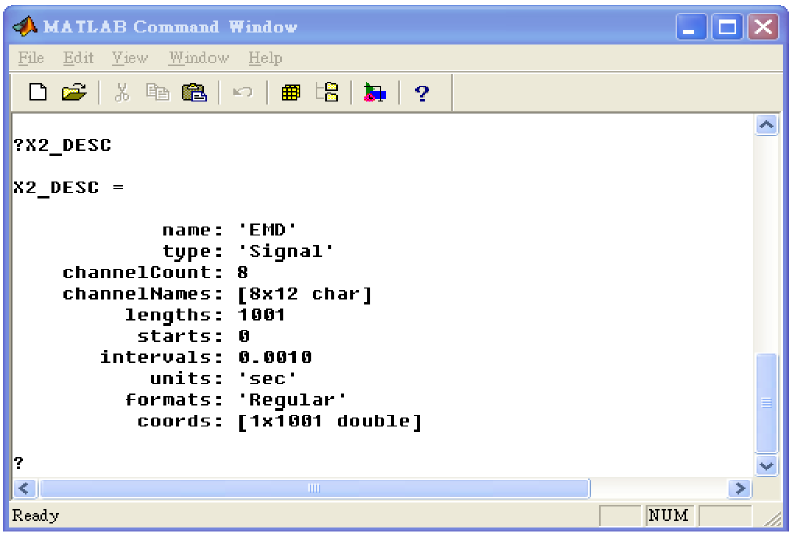
After typing in the command, press ! directly or turn off the editor, the DoMatlab will start to run. As shown below, the two signals overlap completely.



Since the output signal Y is not defined, the error message is shown.

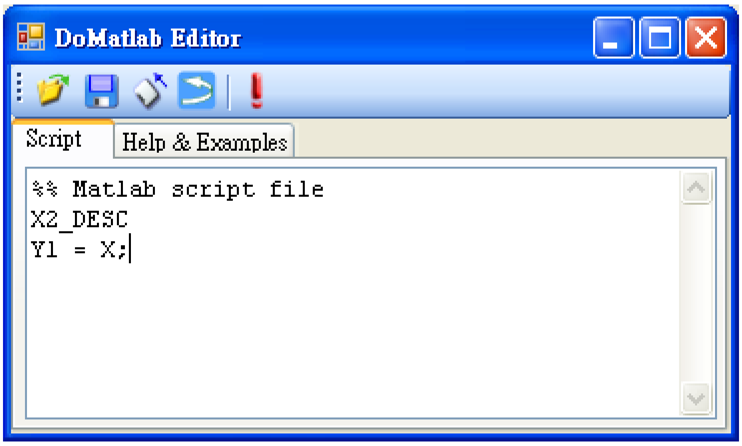


3. To get the basic information of the input signal, enter “X2_DESC” in the command window and press Enter, the information of input signal X2 is displayed.

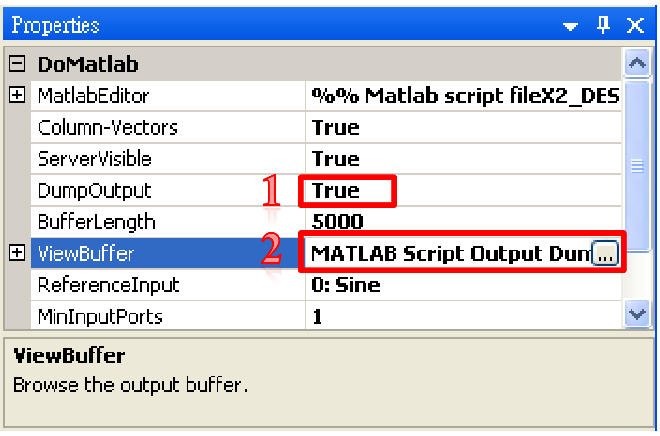


4. The information in step 3 can also be obtained by using the property of

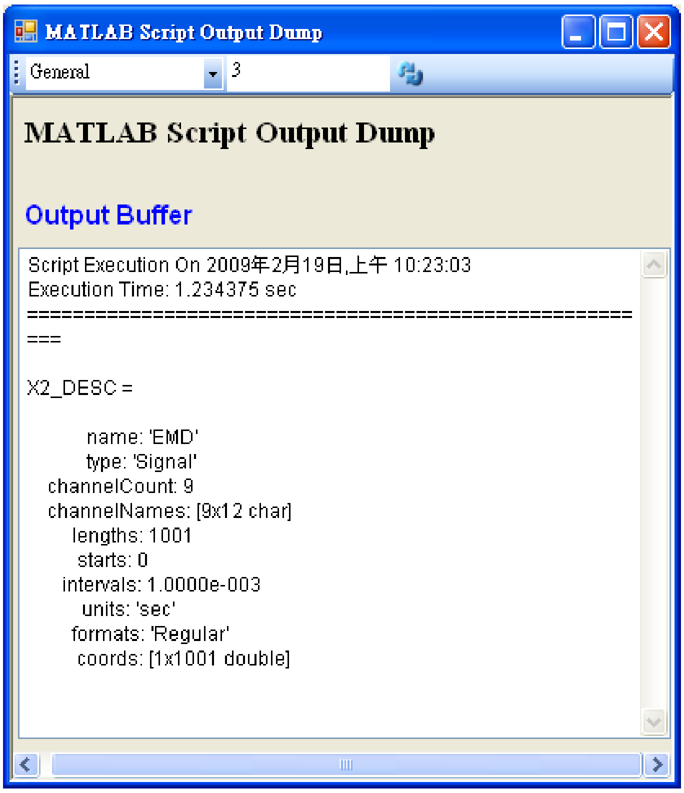
DumpOutput in DoMatlab. First, as shown in the figure below, change the content in DoMatlab Editor to X2_DESC and preserve Y=X1 to define an output signal.



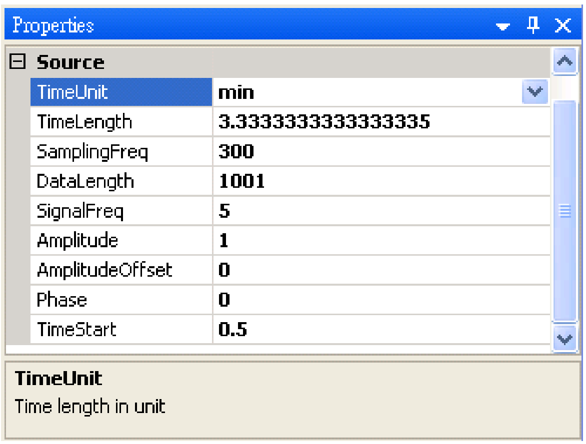
Turn of the editor and change *Properties/DumpOutput* to True. After changing the DumpOutput, DoMatlabe would save all information shown in the command window.



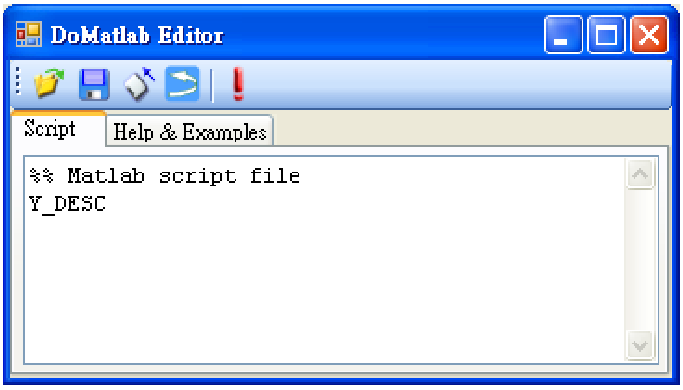
DumpOutput records can be seen in *Properties/*Viewbuffer. Therefore, if any errors occur during the DoMatlab operation, the error messages generated in Matlab are still available.



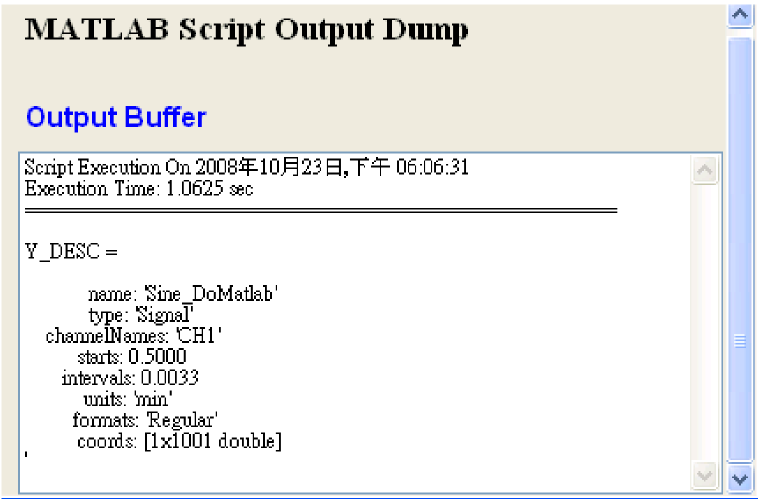
5. The meaning of properties in *Reference Input* is shown below. Go back to *Network*, click Sine and change the *Properties/Time Unit* to minute, the SamplingFreq to 300 and the TimeStart to 0.5.



Then, re-open the DoMatlab Editor and type in Y_DESC. After closing the editor, use the ViewBuffer to see the messages output from Matlab.



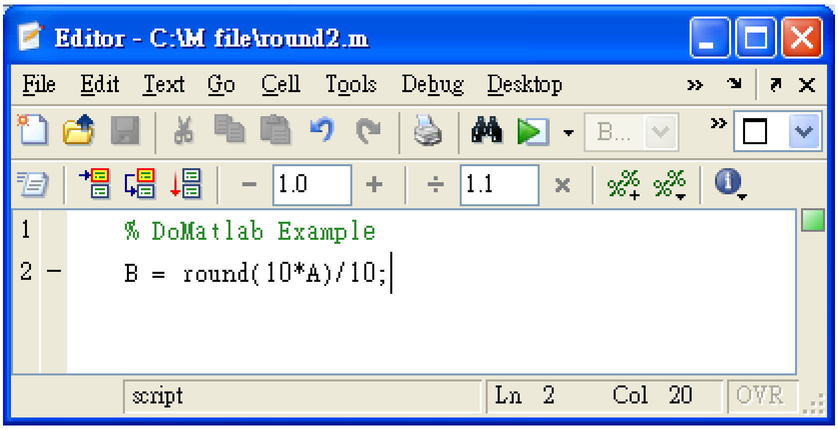
It can be seen that some contents of Y_DESC have been set following the *ReferencInput* although no output signal, i.e. variable Y, has been set in Matlab Editor. As shown below, Y_DESC.starts, Y_DESC.intervals, Y_DESC.units, Y_DESC.coords are all identical to those in ReferenceInput (Sine).



Note that the operation of ReferenceInput in DoMatlab is a little different from that in other module components, such as Merge to Multi-Channel, Merge to Complex, and Math. In DoMatlab, ReferenceInput is used as time-axis coordinate for default output signals while other components would directly replace the time axis coordinates of all input signals with the ones set by ReferenceInput.

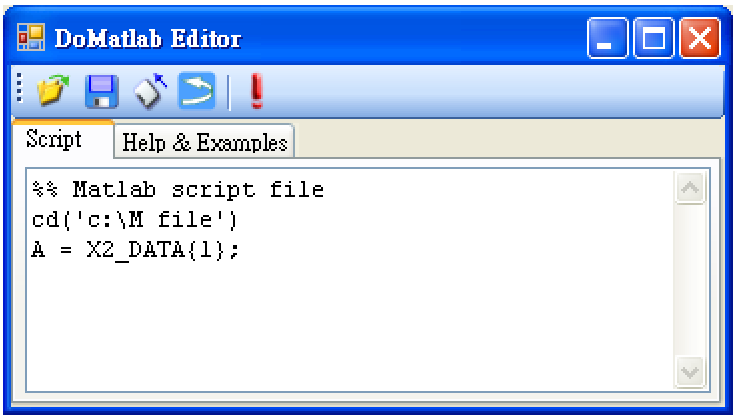
Use DoMatlab to call existing m-files

DoMatlab can run an existing Matlab m-file. Assume there is a m-file, round2.m, which is saved in “C:\M file”. Its content is B=round(10*A)/10 which rounds off A to 1 decimal place.

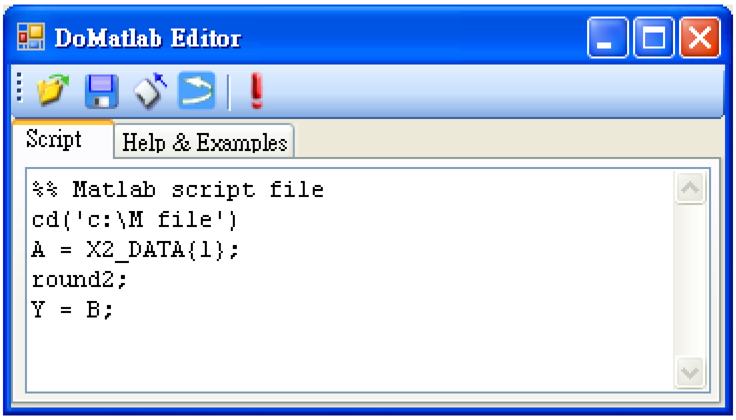


Now, substitute the 1st channel of X2 with A for calculation. The procedures are listed below.

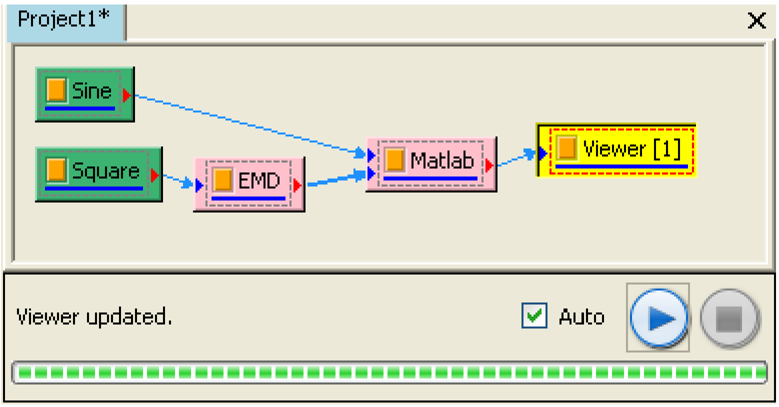
1. First, type in cd(‘C:\M file’) in the Editor, change the current directory to “C:\M file”, and set the 1st channel of X2_DATA as A.

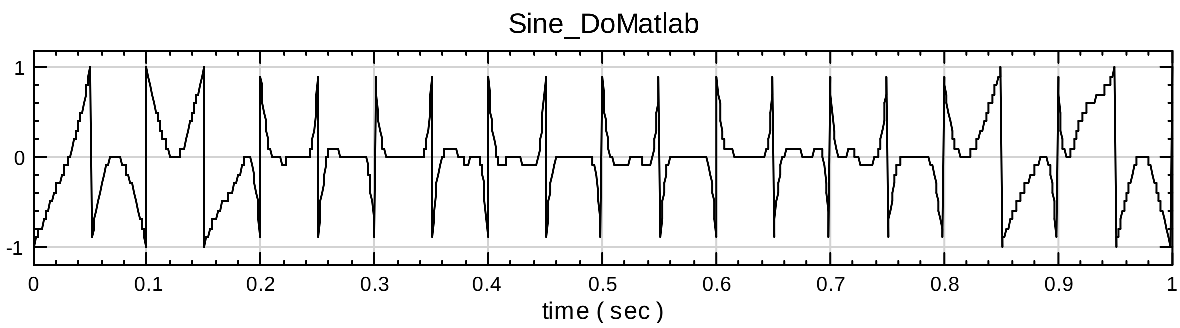


2. Enter round2 in *DoMatlab Editor* to run this m-file. Notice that because the output variable of round2 is B while the output variable of DoMatlab is Y, the B needs to be substituted into Y.



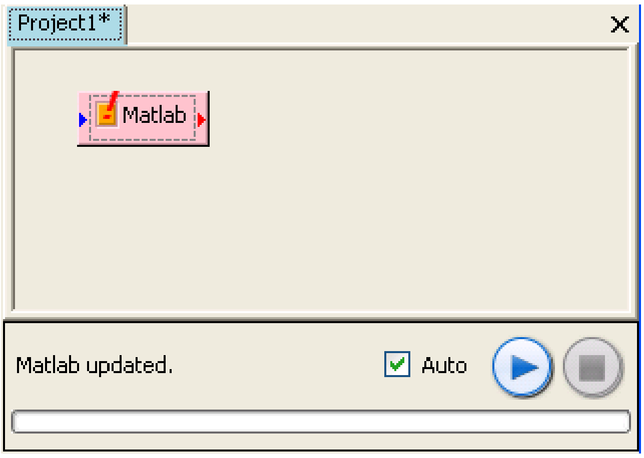
3. Close Editor and use *Viewer→Channel Viewer* to plot the DoMatlab result. The rounding result of the data in the EEMD 1st channel can be observed.

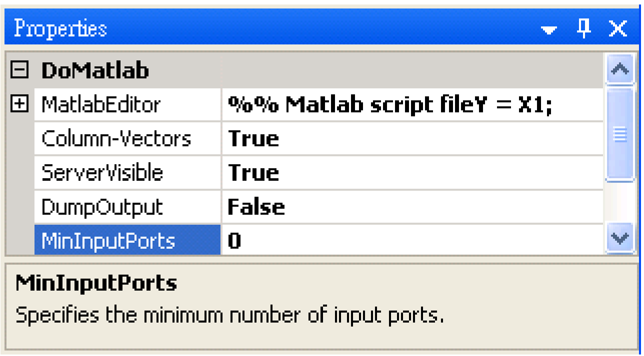


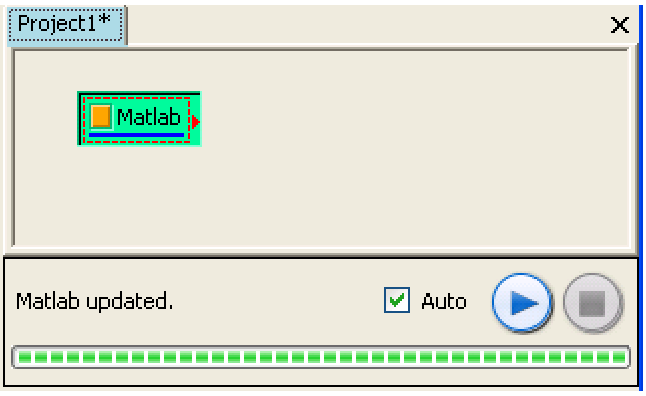


Use DoMatlab to generate new Source

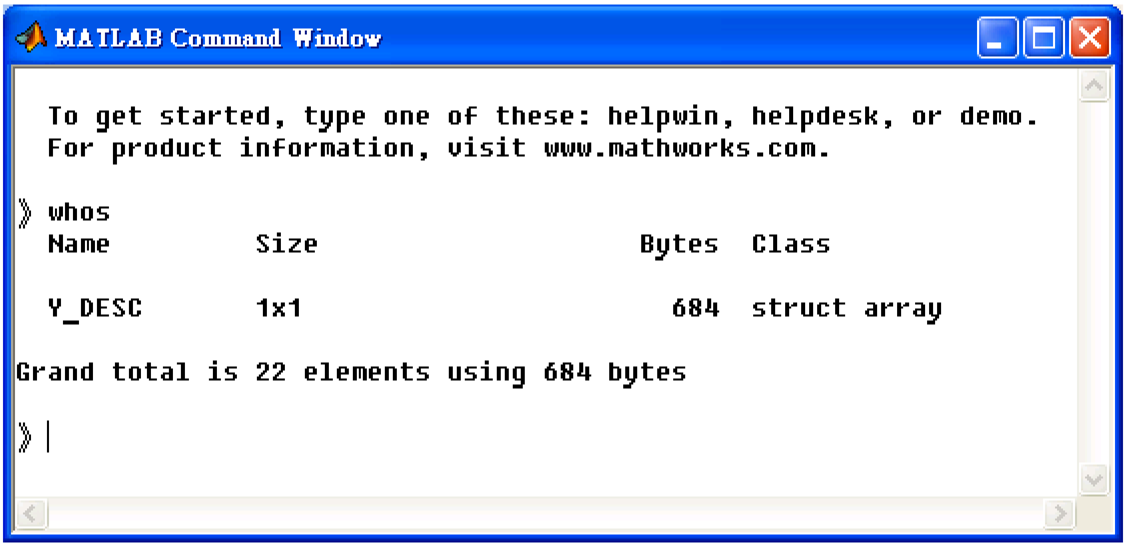
1. Use Computer→Mathematics→ DoMatlab to create a DoMatlab SFO, and change the *Properties*/AsSource to True. This makes the DoMatlab changed from a calculation component to a signal source component. And its color is changed from red to turquoise. The blue input triangle also disappears.



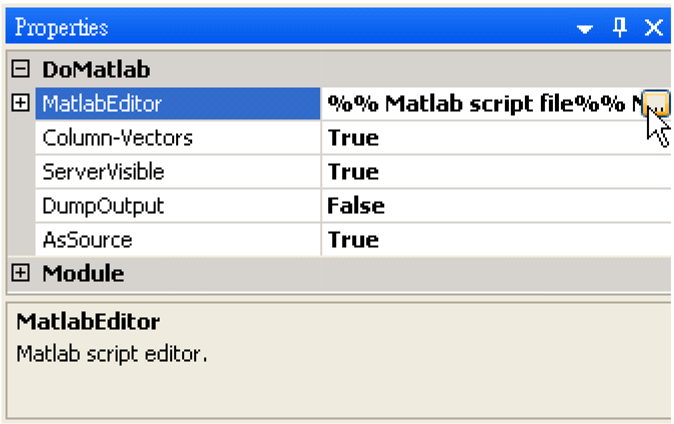




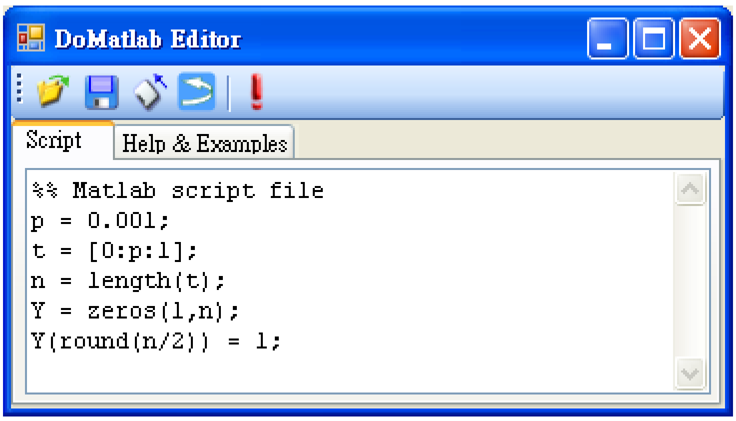
2. In the Matlab Command Window, type in *whos* to search for variables. The result is shown below. Since there is no input data and only a variable of Y_DESC exists, the variable Y needs to be created first and then the setting of data format needs to be filled in Y_DESC.



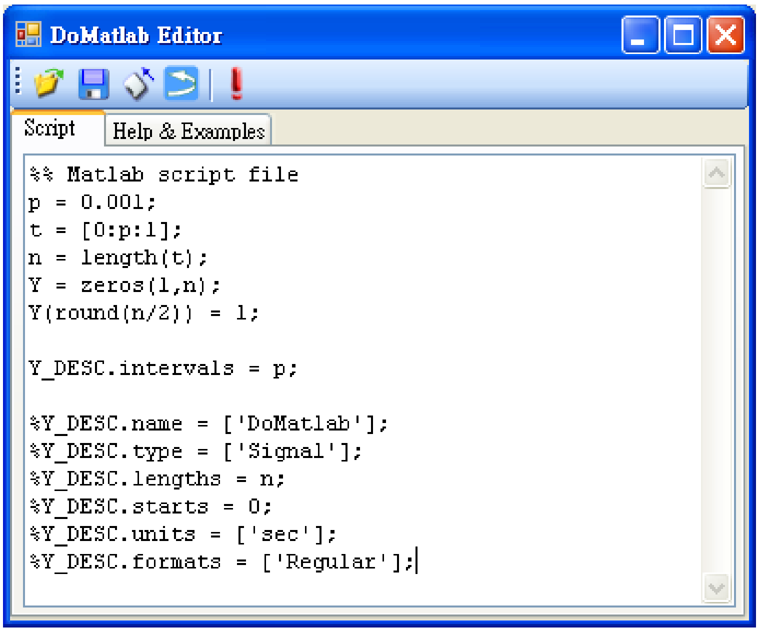
3. Next, use *Properties/*MatlabEditor to create a signal. By setting Y_DESC, DoMatlab can generate signals in all formats. This step shows how to create an impulse signal. Open a new Project and repeat step 1 to create *DoMatlab source* and then enter the editor page by *Properties/*MatlabEditor.



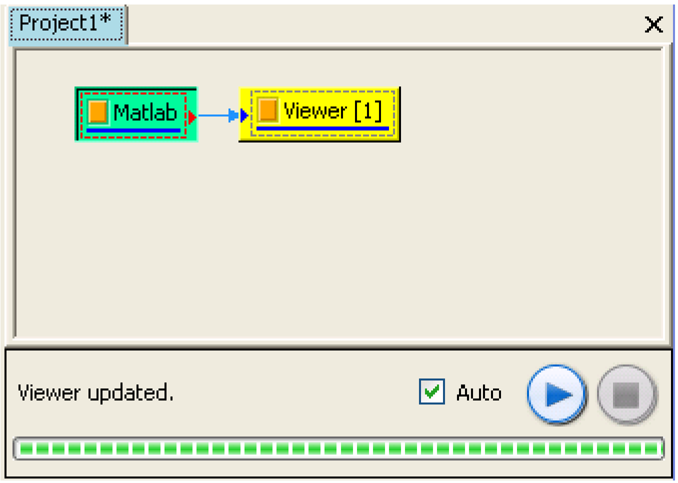
In the editor window, type the codes to generate an Impulse signal. The codes are listed below.

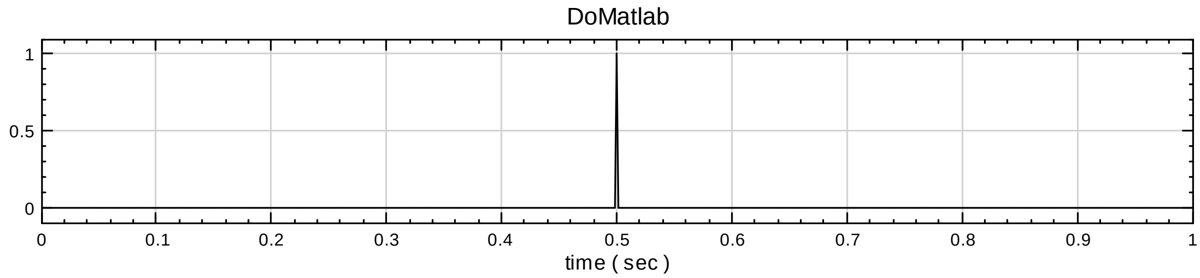


Set variable p=0.001 as the sampling period, variable t is the corresponding time of every signal data, variable n is the signal length, variable Y is the output signal, all data point are pre-set to 0, and set the approximate middle point of the data to 1. Thus, the numerical part of the output signal Y has been created. Next, the time axis variable Y_DESC needs to be set. The codes are shown below.

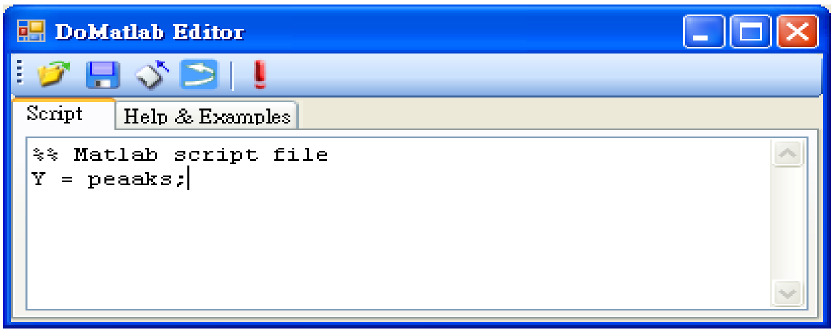


For user convenience, DoMatlab has a pre-set type of output signal as Signal: the start time (Starts) is 0, the time unit (Units) is in sec, the discrete format (Formats) is Regular. All these default values are in the comments, as shown in the figure above. The users only need to set the sampling period, Y_DESC.intervals to obtain Y output. Now, close the Matlab Editor and go back to Network. Connect *Channel Viewer* to DoMatlab and it can be seen that a group of Impulse signals are generated.

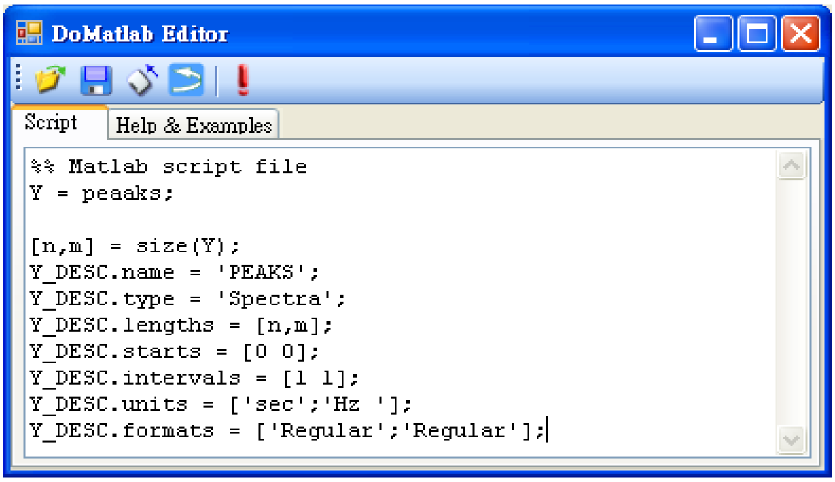




4. In this step, let's use the Matlab internal function, *peaks*, to create a matrix of 49 x 49 and change its signal format to Spectra output. Open a new Project, create a DoMatlab signal source following the step 2 and then open *Properties/*MatlabEditor. First, create a peaks as output signal Y.



Then, define all fields of Y_DESC as shown in the figure below.



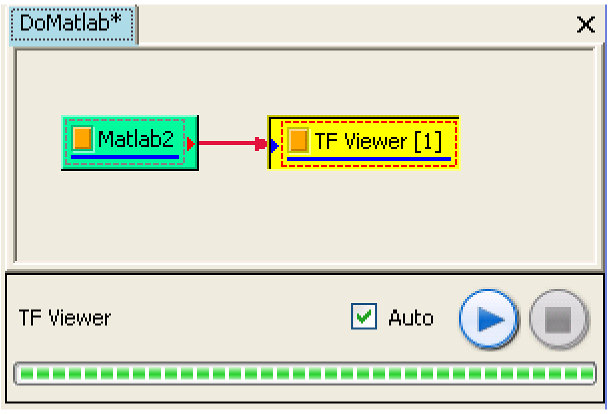
Item	Code	Comment
name of output data	Y_DESC.name = 'PEAKS';	This signal is named PEAKS
Type of output signal	Y_DESC.type = 'Spectra';	The signal type is set as Spectra

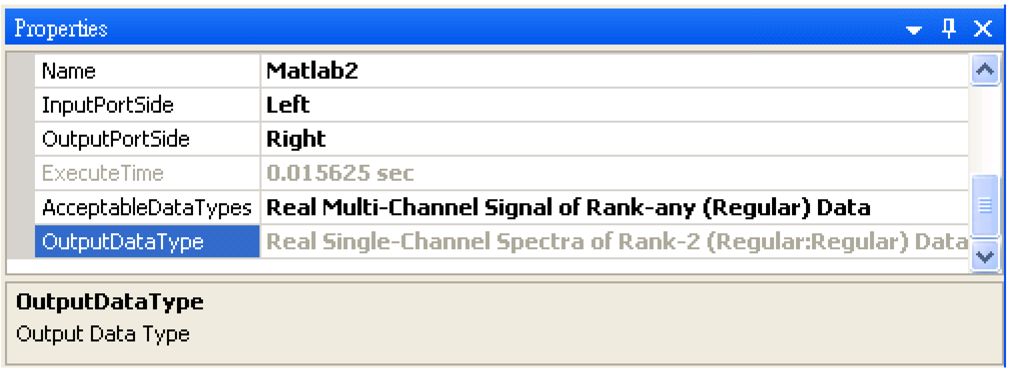
Item	Code	Comment
The number of discrete points in X-axis, Y-axis	Y_DESC.lengths=[n,m];	Because Y is a 2-dimensional array, the lengths of row and column need to be set in order. Corresponding to the axis definition in Spectra, row represents time, i.e., x-axis, column represents frequency, i.e., y-axis
The starting point of x- and y-axis	Y_DESC.starts=[0 0];	Set the starting points of two axis to zero
The interval in x- and y-axis	Y_DESC.intervals =[1 1];	Set the intervals in both axis as 1
The unit of x- and y-axis	Y_DESC.units = [‘sec’;’Hz ’];	Set the unit of x-axis as second. Set the unit of y-axis as Hz.
The discrete type of x- and y-axis	Y_DESC.formats=[‘Regular’;’ Regular’];	Set the discrete type in both axises as equidistant Regular.

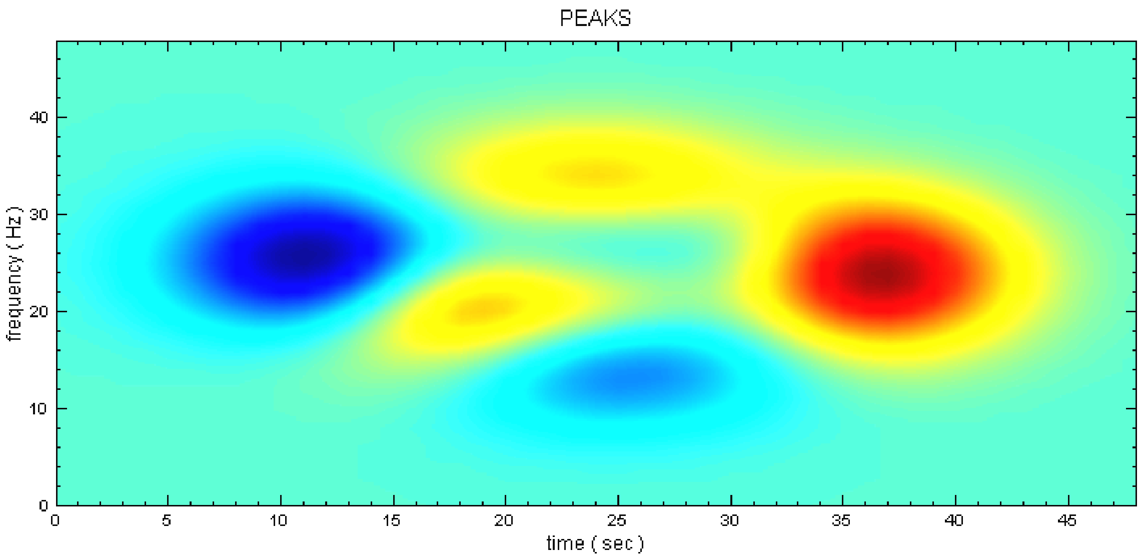
Note: in this table, the relationship between x-, y-axis and the row, column in the array is based the assumption that Column-wise is set as True

The property of Y_DESC.coords only needs to be set when the time axis format is Indexed and it is not required in this example.

Run DoMatlab after closing DoMatlab Editor. Go back to Network window and click DoMatlab to see the *Properties/*OutputDataType. The output signal format is *Real Single-Channel Spectra of Rank-2 (Regular:Regular) Data* and it could be connected to *Viewer→Time-Frequency Viewer* to plot the time-frequency diagram.







Related Functions

Viewer, Differentiate, Integrate, Source

3.4 Statistics (Professional Only)

This group of modules provides statistical calculation for signals. The components are:

Basic Statistics: basice statistics for the signal, such as maximum, minimum, average, standard deviation.

Covariance Matrix: calculate covariance matrix among signals.

Correlation Matrix: calculate correlation maxtrix among signals.

Equiphase Statistics: calculate equiphase statistics for the signal.

Kernel Smoothing Density: calculate the probability density function of the signal using special kernel function and smooth the result.

Orthogonality Matrix: calculate the orthogonality matrix among multiple signals.

Quartiles and Quantiles: calculate different quartiles and quantiles of the signal.

Rolling Statistics: calculate rolling statistics for the signal.

Hypothesis Test: build the hypothesis for the signal, select the test method, and validate hypothesis.

3.4.1 Basic statistics

Basic statistics is a quick way to get basic statistical values for a signal.

Introduction

Let the signal series be $X = \{x_0, x_1..x_{N-1}\}$, N is the length of the signal. It is not limited to time series. Basic Statistics gives following calculation.

Statistics	Formula	Description
Sum	$\sum_{i=0}^{N-1} x_i$	Sumation of all the elements
Min	-	Minimum value
Max	-	Maximum value
Mean	$\frac{1}{N} \sum_{i=0}^{N-1} x_i$	Average value
Geometric Mean	$(x_0 \cdot x_1 \cdot \cdot x_{N-1})^{\frac{1}{N}}$ $x_i > 0,\ for\ all\ i$	Geometric mean, mainly used in exponential change series, such population growth, raio calculation.
Harmonic Mean	$\frac{N}{\sum_{i=0}^{N-1} \frac{1}{x_i}}$ $x_i > 0,\ for\ all\ i$	Harmonic average, maily used in average speed calculation.
Trimmed Mean	-	Average with removing top and bottom percentile. Set a percentile value (%), sort X series, remove top and bottom

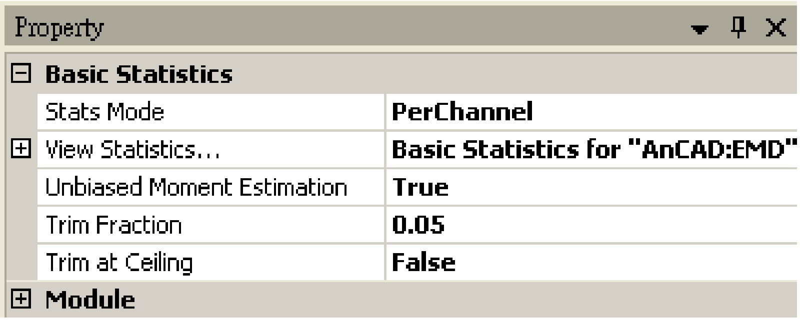
		percentile, average the remaining elements. This removes the effect due to outliers.
Median	-	Median Value
StdDev	$\sigma = \left(\frac{1}{N} \sum_{i=0}^{N-1} (x_i - \bar{x})^2 \right)^{\frac{1}{2}}$	Standard deviation, estimate the deviation from the average. This is Biased Moment Estimation.
	$s = \left(\frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \bar{x})^2 \right)^{\frac{1}{2}}$	Assume it is the true Sample. This is Unbiased Moment Estimatiom.
Variance	$\sigma^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - \bar{x})^2$	Variance (square of StdDev). This is Biased Moment Estimation.
	$s^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \bar{x})^2$	This is Unbiased Moment Estimation.
Coefficient of Variation	$\frac{\sigma}{\bar{x}} \times 100\%$	The ratio between variance and average. It used to show the discrete degree. This is Biased Moment Estimation.
	$\frac{s}{\bar{x}} \times 100\%$	This is Unbiased Moment Estimation.

Skewness	$$g_1 = \frac{E[(x-\bar{x})^3]}{E[(x-\bar{x})^2]^{\frac{3}{2}}} = \frac{\frac{1}{N}\sum_{i=0}^{N-1}(x_i-\bar{x})^3}{\left(\frac{1}{N}\sum_{i=0}^{N-1}(x_i-\bar{x})^2\right)^{\frac{3}{2}}}$$	A measure of the asymmetry of the probability distribution. This is Biased Moment Estimation.
	$$G_1 = \frac{\sqrt{N(N-1)}}{N-2} \cdot g_1$$	This is Unbiased Moment Estimatiom.
Kurtosis	$$g_2 = \frac{\sum_{i=0}^{N-1}(x_i-\bar{x})^4}{\left[\frac{1}{N}\sum_{i=0}^{N-1}(x_i-\bar{x})^2\right]^2} - 3$$	A measure of the “peakedness” of the probability districution. Higer Kurtosis means more of the variance is the result of infrequent extreme deviations. This is Biased Moment Estimation.
	$$G_2 = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \cdot \frac{\sum_{i=0}^{N-1}(x_i-\bar{x})^4}{s^4} - \frac{3(N-1)^2}{(N-2)(N-3)},$$	This is Unbiased Moment Estimatiom.
Semivariance	$$\frac{\sum_{i=0}^{N-1}[(x_i-\bar{x}) \cdot b_i]^2}{N}$$ $$b_i = \begin{cases} 1, x_i < \bar{x} \\ 0, x_i > \bar{x} \end{cases}$$	Degree of spatial dependences of a spatial stochastic process. The default threshold is the average value. This is Biased Moment Estimation.
	$$\frac{\sum_{i=0}^{N-1}[(x_i-\bar{x}) \cdot b_i]^2}{N-1}$$	This is Unbiased Moment Estimatiom.

	$b_i=\begin{cases}1, x_i<\bar{x}\\ 0, x_i>\bar{x}\end{cases}$	
Semi Standard Deviation	$$\sqrt{\frac{\sum_{i=0}^{N-1}[(x_i-\bar{x})\cdot b_i]^2}{N}}$$ $b_i=\begin{cases}1, x_i<\bar{x}\\ 0, x_i>\bar{x}\end{cases}$	This is Biased Moment Estimation.
	$$\sqrt{\frac{\sum_{i=0}^{N-1}[(x_i-\bar{x})\cdot b_i]^2}{N-1}}$$ $b_i=\begin{cases}1, x_i<\bar{x}\\ 0, x_i>\bar{x}\end{cases}$	This is Unbiased Moment Estimatiom.

Properties

This module accepts input of Signal which could be real number or complex number, single channel or multi-channel, regular.



Property	
Basic Statistics	
Stats Mode	PerChannel
View Statistics...	Basic Statistics for "AnCAD:EMD"
Unbiased Moment Estimation	True
Trim Fraction	0.05
Trim at Ceiling	False
Module	

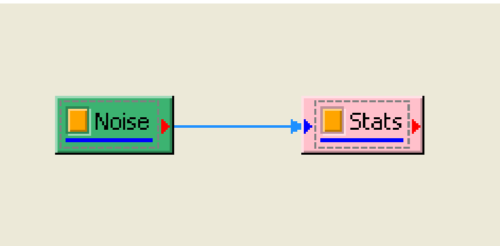
Property Name	**Property Definition**	**Default Value**
Stats Mode	If the input is a multi-channel signal, this option is activated: PerChannel or AcrossChannel. For PerChannel setting, the result is a 15 * n matrix, where n is the channel number of the input, 15 is the number of calculated statistical value. The output is in Indexed format.	PerChannel

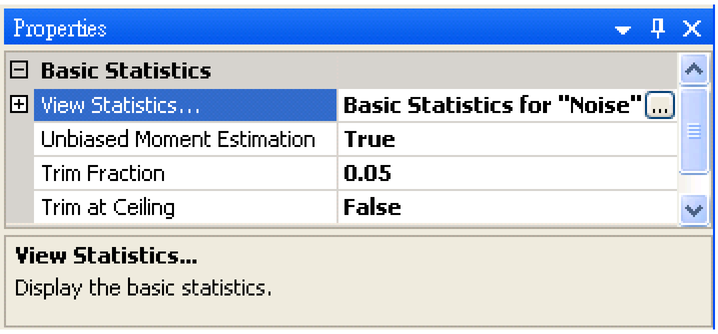
	For AcrossChannel setting, calculate 15 statistical values for the data of all channels at the same time point. The output signal contains 15 channels and its length is the same as the input signal.	
View Statistics…	Open Reporter window to view the result.	None
Unbiased Moment Estimation	If the input is Sample, it is True (Unbiased Moment Estimation) for calculating the statistics of Population. If the input is Population, it is False (biased Moment Estimation) for calculating the statistics of Population.	True
Trimmed Fraction	The percentile of the top and bottom segments to be removed.	0.05
Trimmed at Ceiling	If the position of specific percentile is not an integer, include the previous point (False) or the next point (True) to be removed.	False

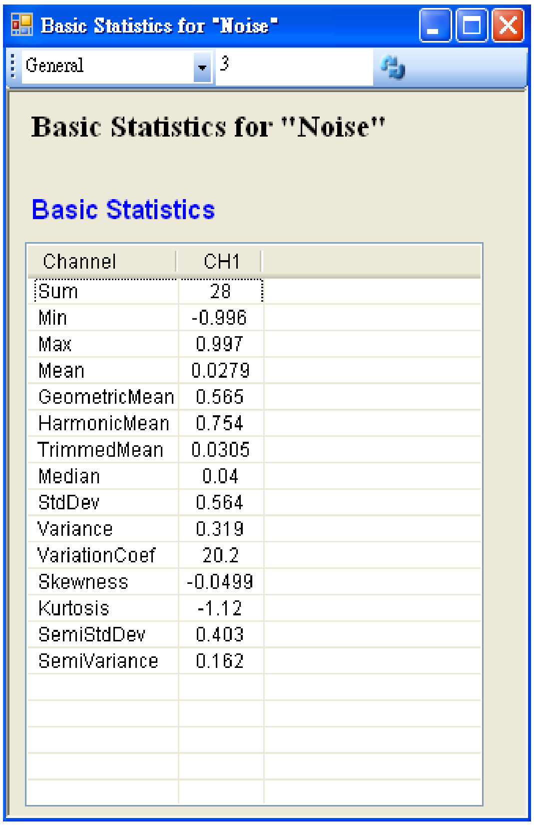
Example

Calculate the basic statistics of white noise and square wave:

Create a noise using *Source / Noise* (default is white noise) in Network panel, connect this SFO to *Compute / Statistics / Basic Statistics*. Select *Properties / View Statistics…* of Basic Statistics to view result.

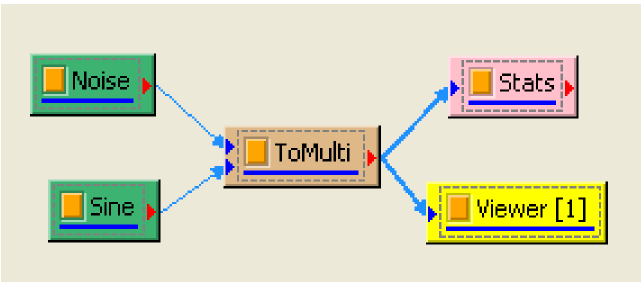


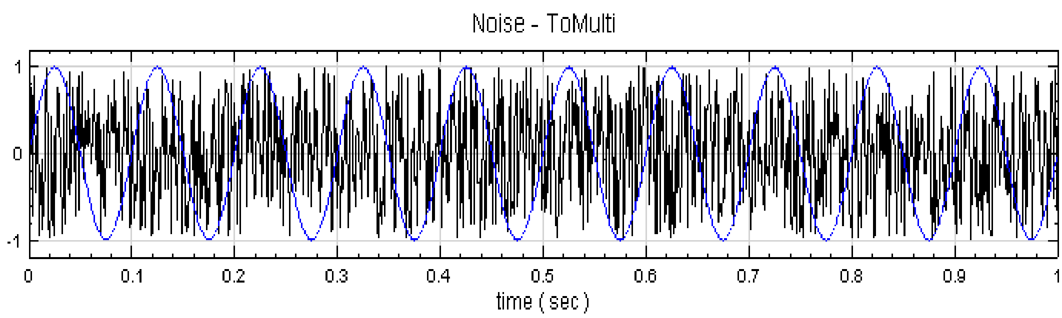


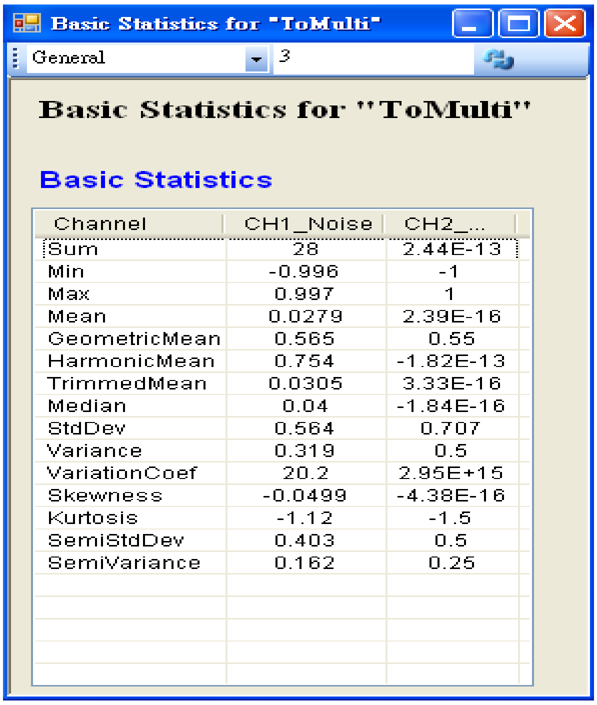


Channel	CH1
Sum	28
Min	-0.996
Max	0.997
Mean	0.0279
GeometricMean	0.565
HarmonicMean	0.754
TrimmedMean	0.0305
Median	0.04
StdDev	0.564
Variance	0.319
VariationCoef	20.2
Skewness	-0.0499
Kurtosis	-1.12
SemiStdDev	0.403
SemiVariance	0.162

Add a Sine Wav to the Noise using *Conversion / Merge to multi-channel* and display both signals with *Viewer / Channel viewer.* Then connect to Stats and display the result using *Properties / View Statistics…* in Basic Statistics.

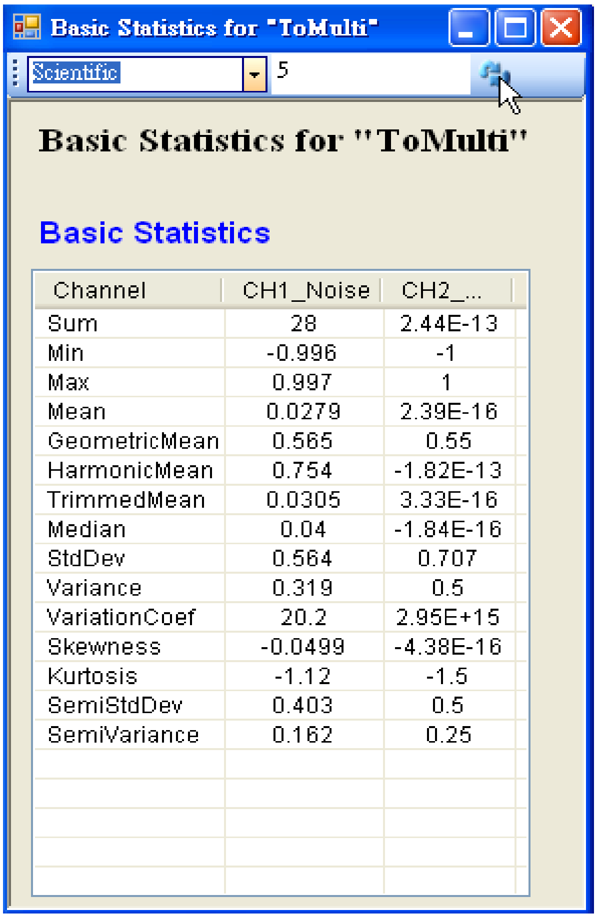






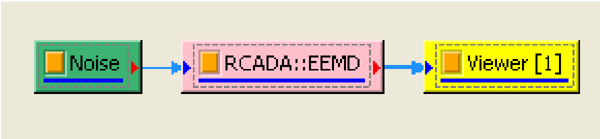
Channel	CH1_Noise	CH2_...
Sum	28	2.44E-13
Min	-0.996	-1
Max	0.997	1
Mean	0.0279	2.39E-16
GeometricMean	0.565	0.55
HarmonicMean	0.754	-1.82E-13
TrimmedMean	0.0305	3.33E-16
Median	0.04	-1.84E-16
StdDev	0.564	0.707
Variance	0.319	0.5
VariationCoef	20.2	2.95E+15
Skewness	-0.0499	-4.38E-16
Kurtosis	-1.12	-1.5
SemiStdDev	0.403	0.5
SemiVariance	0.162	0.25

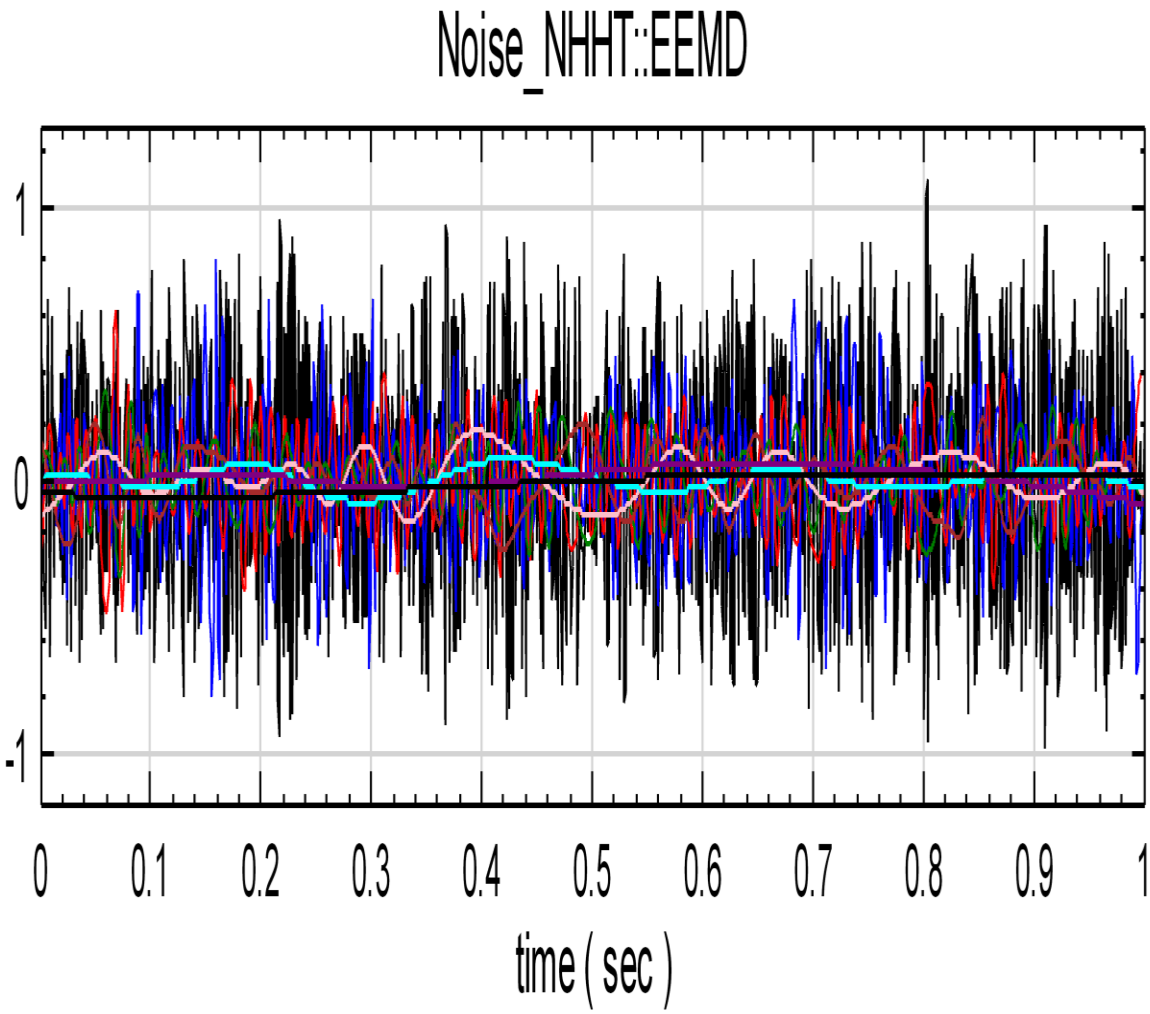
Set the parameter in Basic Statistics toolbox to Scientific and set the digits to 5 after decimal point. Click “Refresh” button to update the result.



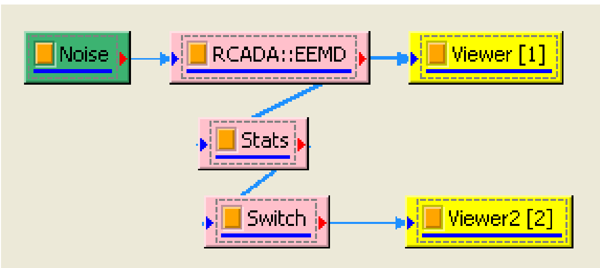
Channel	CH1_Noise	CH2_...
Sum	28	2.44E-13
Min	-0.996	-1
Max	0.997	1
Mean	0.0279	2.39E-16
GeometricMean	0.565	0.55
HarmonicMean	0.754	-1.82E-13
TrimmedMean	0.0305	3.33E-16
Median	0.04	-1.84E-16
StdDev	0.564	0.707
Variance	0.319	0.5
VariationCoef	20.2	2.95E+15
Skewness	-0.0499	-4.38E-16
Kurtosis	-1.12	-1.5
SemiStdDev	0.403	0.5
SemiVariance	0.162	0.25

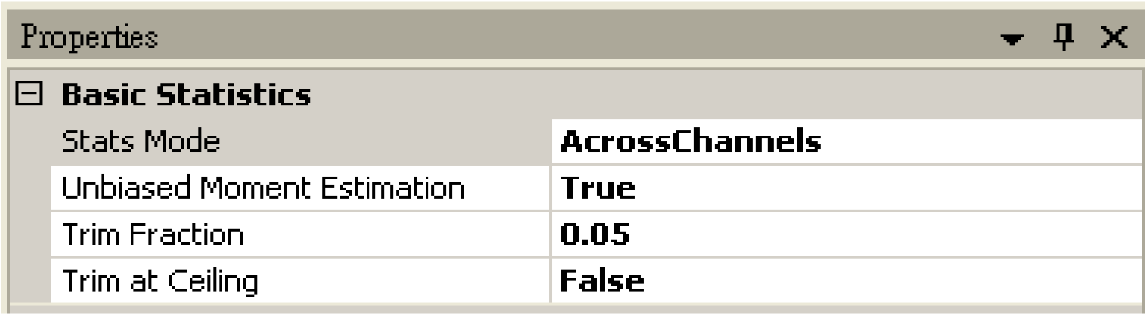
Create another white noise signal using *Source / Noise*, then connect it to *HHT / RCADA EEMD* SFO, and display the result with *Channel Viewer*.





Finally connect *RCADA EEMD* SFO to *Compute / Statistics / Basic Statistics* and set *Properties / Stats Mode* to AcrossChannels. Then display the results using *Compute / Channel / Channel Switch* and *Channel Viewe*. With Channel Switch, different channel can be selected for viewing.





Related Functions

Equiphase Statistics, Rolling Statistics, Merge to Multi.

Reference

Michel Loeve, "Probability Theory", Graduate Texts in Mathematics, Volume 45, 4th edition, Springer-Verlaf, 1977

Joanes, D. N.&Gill, C. A. (1998) Comparing measuresof sample skewness and kurtosis. Journal of the Royal Statistical Society (Series D): The Statistician 47 (1), 183–189.

3.4.2 Covariance Matrix

Covariance is a measure of how much two series change together relative to their average value. If covariance is positive, it means that two series change in the same direction. If covariance is negative, it means that two series change in the opposite direction. If there are multiple series involved, covariance matrix is used to show the covariance between each pair series.

Introduction

Let $X = \{x_0, x_1 .. x_{N-1}\}$, $Y = \{y_0, y_1, ..., y_{N-1}\}$ be two series, the definition of covariance is:

$$cov_{xy}^{(b)} = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x})(y_i - \bar{y})}{N}$$

where $\bar{x}$, $\bar{y}$ are average of each series.

For unbiased moment estimator, the definition of covariance is:

$$cov_{xy}^{(s)} = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

If there is multi-channel series, the total channel number is M, covariance matrix can be represented by

$[C_{lk}] = cov_{lk}$, where l, k are channel number

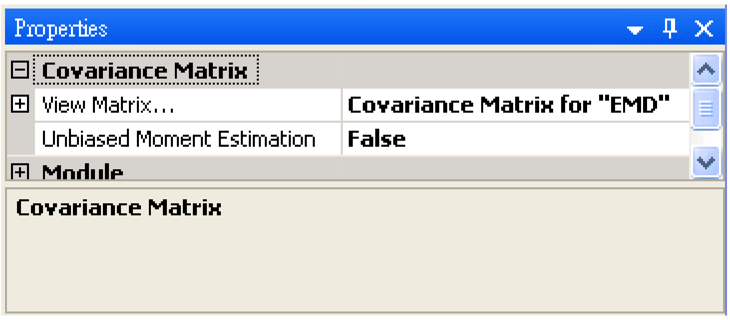
And the Diagonal terms of the matrix are covariance for each channel:

$$cov_{kk}^{(b)} = \frac{\sum_{i=0}^{N-1} (x_i^{(k)} - \bar{x}^{(k)})(y_i^{(k)} - \bar{y}^{(k)})}{N}$$

$$cov_{kk}^{(s)} = \frac{\sum_{i=0}^{N-1} (x_i^{(k)} - \bar{x}^{(k)})(y_i^{(k)} - \bar{y}^{(k)})}{N-1}$$

Properties

This module accepts input of real number, multi-channel, and regular signal. The output is a MxM matrix, where M is the total channel number. And the output is in Indexed format. The result can be viewed in Reporter windown by clicking *Properties / View Matrix…*

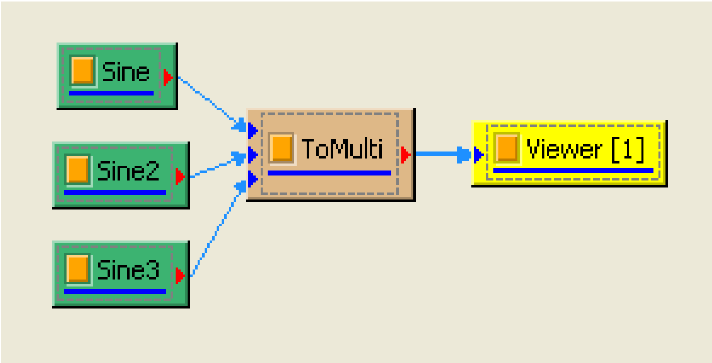


Property Name	Property Definition	Default value
Unbiased Moment Estimation	Calculate the covariance using unbiased moment estimation method	False

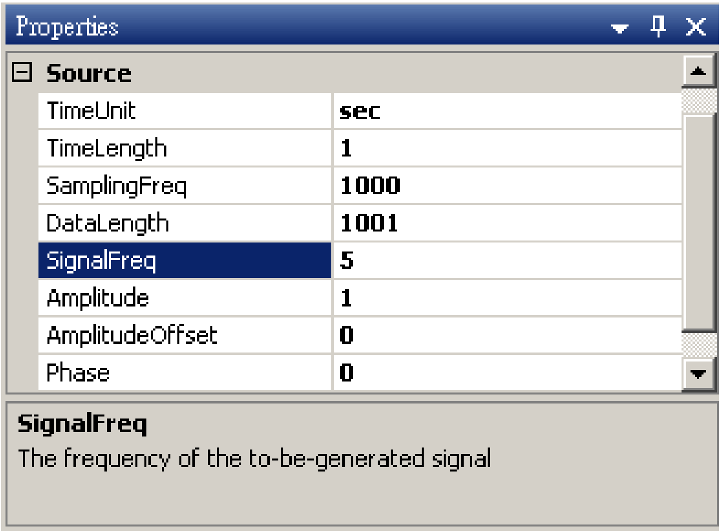
Example

Calculate covariance matrix for sine waves with different phase and frequency:

Create a since wave in Netowrk panel using *Source / Sine Wave* and its default frequency is10Hz. Then add two more sine waves. One is set with *Properties / SignalFreq* = 5Hz and the other is set with *Properties / SignalFreq* = 10Hz and *Properties / Phase* = 180 degree (unit is in Degree). Finally, merge three signals into a Multi-channel signal using *Conversion / Merge to Multi-channel*. In this case, three sine waves are generated: 10Hz, 5Hz, and 10Hz with phase shift of 180 degree. Using *Viewer / Channel Viewer*, the signals can be displayed. Black curve is Sine, blue curve is Sine2, and red curve is Sine3.



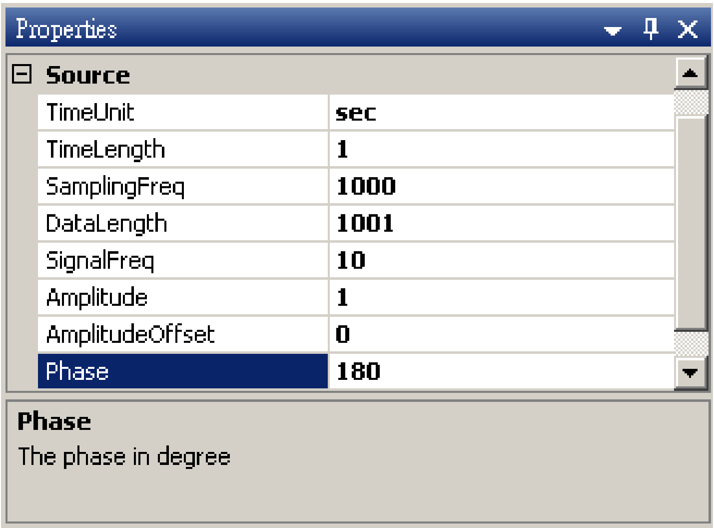
The property of Sine2 is shown here,



TimeUnit	sec
TimeLength	1
SamplingFreq	1000
DataLength	1001
SignalFreq	5
Amplitude	1
AmplitudeOffset	0
Phase	0

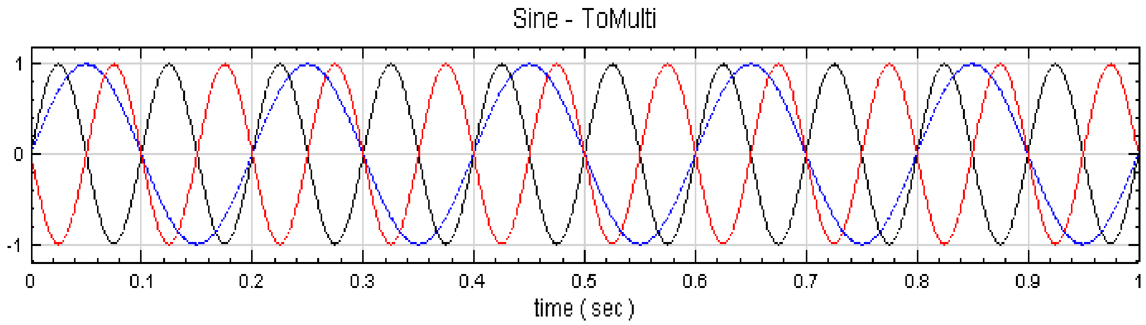
SignalFreq
The frequency of the to-be-generated signal

The property of Sine3 is shown here,

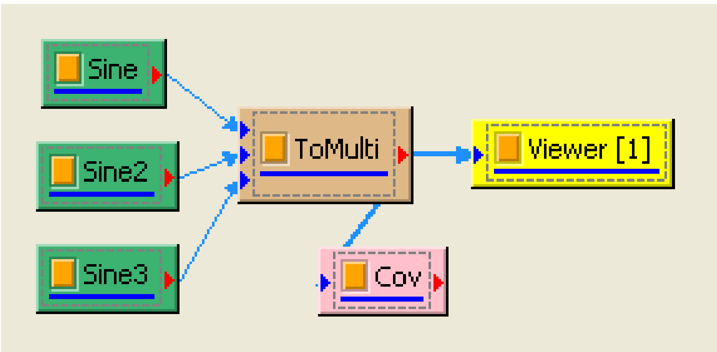


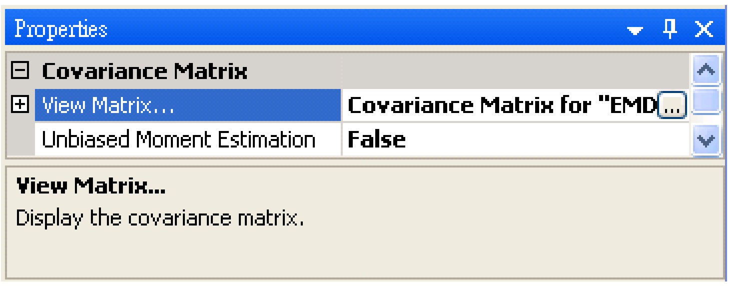
TimeUnit	sec
TimeLength	1
SamplingFreq	1000
DataLength	1001
SignalFreq	10
Amplitude	1
AmplitudeOffset	0
Phase	180

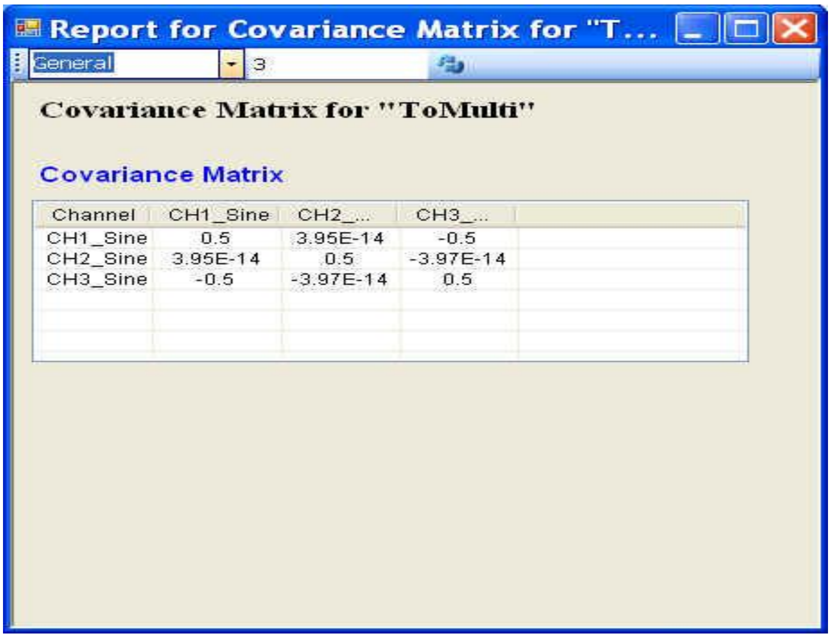
Phase
The phase in degree



Connect ToMulti SFO to *Compute / Statistics / Covariance Matrix* and click *Properties / View Matrix…* to show the result in the pop-up window. The matrix element C_{ii} is self covariance value which is 0.5. C_{12}, C_{21} are covariance between Sine and Sine2. Since the value is very small, it means there is no correlation between these two signals. C_{23}, C_{32} are also very small. It means that there is no correlation between Sine2 and Sine3. Signals Sine and Since3 mirror along X axis and the covariance is about -0.5, it shows that two signals are negative correlated.






Report for Covariance Matrix for "T...

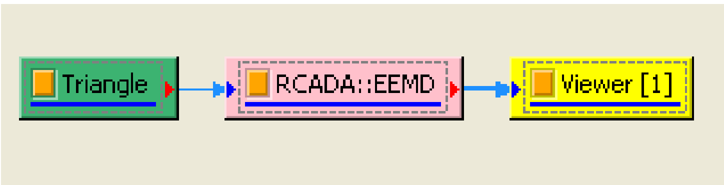
General 3

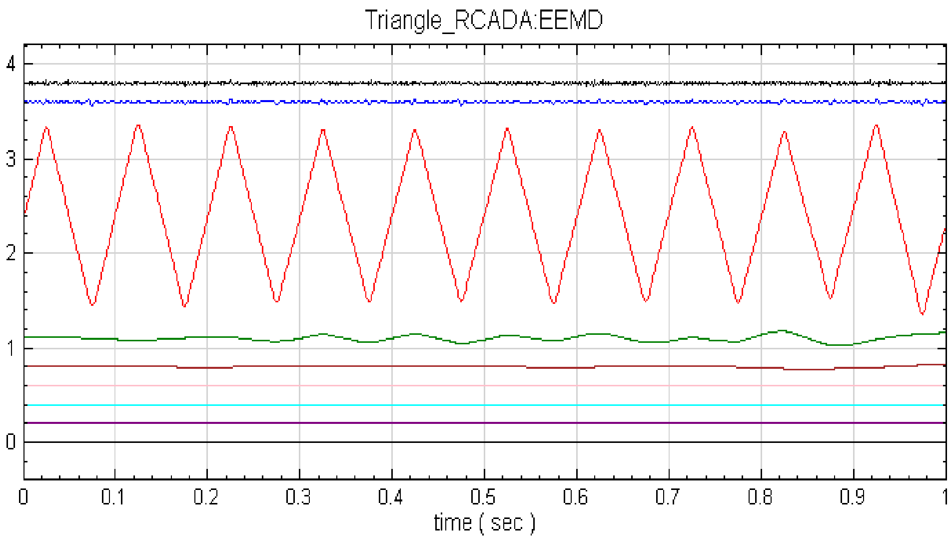
Covariance Matrix for "ToMulti"

Covariance Matrix

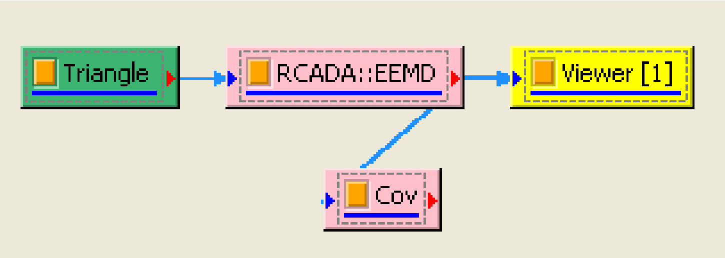
Channel	CH1_Sine	CH2_...	CH3_...
CH1_Sine	0.5	3.95E-14	-0.5
CH2_Sine	3.95E-14	0.5	-3.97E-14
CH3_Sine	-0.5	-3.97E-14	0.5

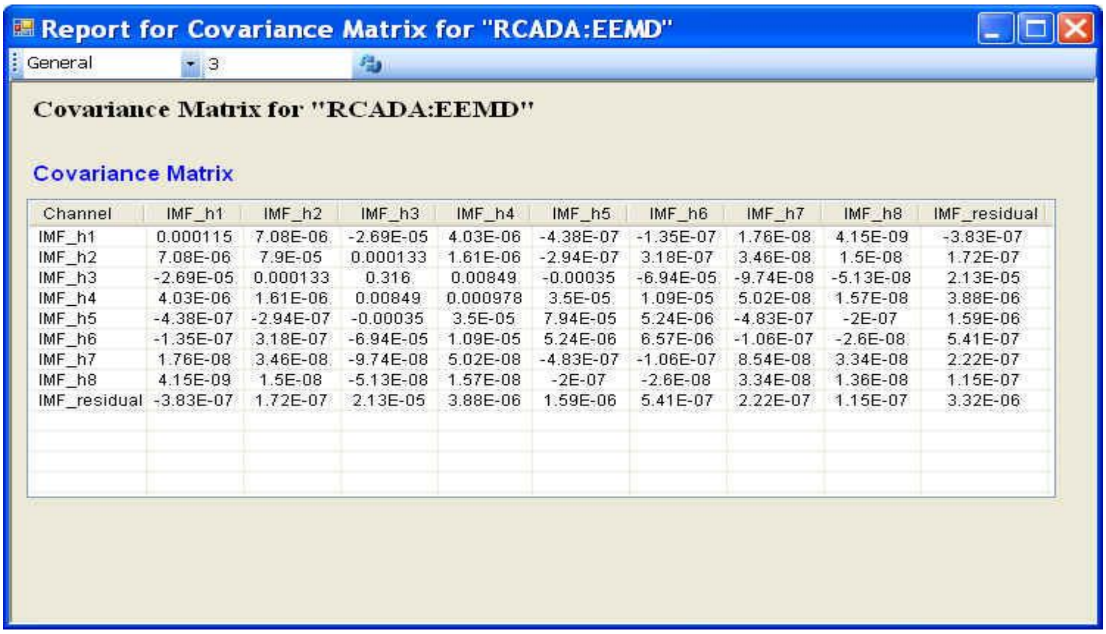
Create a new signal with *Source / Triangle Wave* and connec it to *Compute / HHT / RCADAEEMD* to calculate its IMFs. The results are displayed using *Channel Viewer.* Set *Properties / Multi-channel Display* in the view to List and set Viewer Height to 350. There are 9 channels in the signal.





Connect *RCADA EEMD* SFO to *Compute / Statistics / Covariance Matrix* for calculating covariance matrix among IMFs.





Covariance Matrix for "RCADA:EEMD"

Covariance Matrix

Channel	IMF_h1	IMF_h2	IMF_h3	IMF_h4	IMF_h5	IMF_h6	IMF_h7	IMF_h8	IMF_residual
IMF_h1	0.000115	7.08E-06	-2.69E-05	4.03E-06	-4.38E-07	-1.35E-07	1.76E-08	4.15E-09	-3.83E-07
IMF_h2	7.08E-06	7.9E-05	0.000133	1.61E-06	-2.94E-07	3.18E-07	3.46E-08	1.5E-08	1.72E-07
IMF_h3	-2.69E-05	0.000133	0.316	0.00849	-0.00035	-6.94E-05	-9.74E-08	-5.13E-08	2.13E-05
IMF_h4	4.03E-06	1.61E-06	0.00849	0.000978	3.5E-05	1.09E-05	5.02E-08	1.57E-08	3.88E-06
IMF_h5	-4.38E-07	-2.94E-07	-0.00035	3.5E-05	7.94E-05	5.24E-06	-4.83E-07	-2E-07	1.59E-06
IMF_h6	-1.35E-07	3.18E-07	-6.94E-05	1.09E-05	5.24E-06	6.57E-06	-1.06E-07	-2.6E-08	5.41E-07
IMF_h7	1.76E-08	3.46E-08	-9.74E-08	5.02E-08	-4.83E-07	-1.06E-07	8.54E-08	3.34E-08	2.22E-07
IMF_h8	4.15E-09	1.5E-08	-5.13E-08	1.57E-08	-2E-07	-2.6E-08	3.34E-08	1.36E-08	1.15E-07
IMF_residual	-3.83E-07	1.72E-07	2.13E-05	3.88E-06	1.59E-06	5.41E-07	2.22E-07	1.15E-07	3.32E-06

Covariance is a measure of how much two series change together. If it's positive, the change is in the same direction. If it's negative, the change is in the opposite direction. The elements along the diagonal line of the matrix are covariances between channels.

Related Functions

Correlation Matrix, Orthogonality Matrix, Merge To Multi-Channel, RCADA EMD.

Reference

N.G. van Kampen, Stochastic processes in physics and chemistry. New York: North-Holland, 1981.

3.4.3 Correlation Matrix

Correlation Coefficient is normalized covariance. If there are multiple series, the correlation coefficient matrix consists of the pair-wise correlation coefficient.

Introduction

Let $X=\{x_0, x_1..x_{N-1}\}$, $Y=\{y_0, y_1, ..., y_{N-1}\}$ be two series, the definition of Correlation Coefficient is:

$$\rho_{xy} = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x})(y_i - \bar{y})}{N\sigma_x \sigma_y}$$

where $\bar{x}$, $\bar{y}$ are average of each series, σ_x, σ_y are standard deviation.

For unbiased moment estimator, the definition of Correlation Coefficient is:

$$r_{xy} = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x})(y_i - \bar{y})}{(N-1)s_x s_y}$$

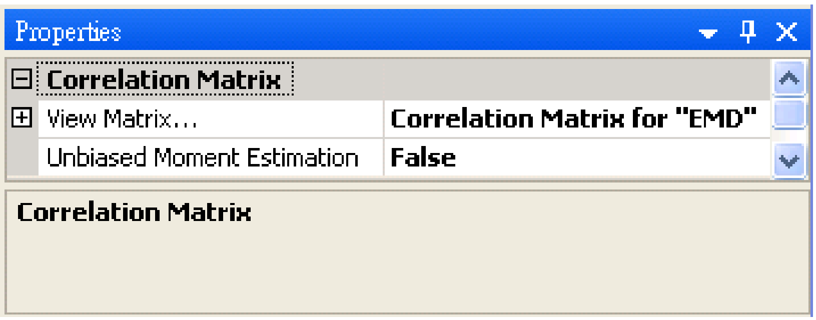
where r_{xy} is the correlation coefficient of the two series, s_x , s_y are standard deviation of each series.

Correlation coefficient is the ratio between covariance and standard deviation of two series. If there is a multi-channel series, the total channel number is M, correlation coefficient matrix can be represented by

$[R_{lk}] = r_{lk}$, where l, k are channel number

Properties

This module accepts input of real number, multi-channel, and regular signal. The output is a MxM matrix, where M is the total channel number. And the output is in Indexed format. The result can be viewed in Reporter windown by clicking *Properties / View Matrix…*

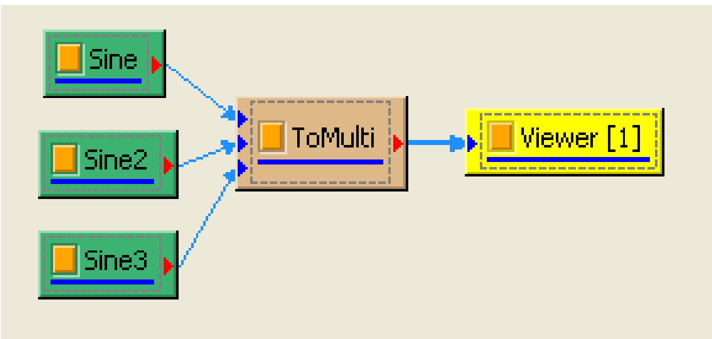


Property Name	Property Definition	Default value
Unbiased Moment Estimation	Calculate the covariance using unbiased moment estimation method	False

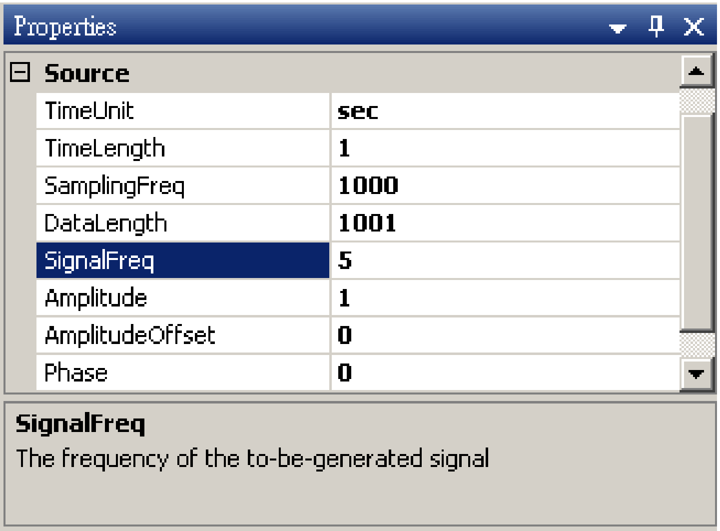
Example

Calculate covariance matrix for sine waves with different phase and frequency:

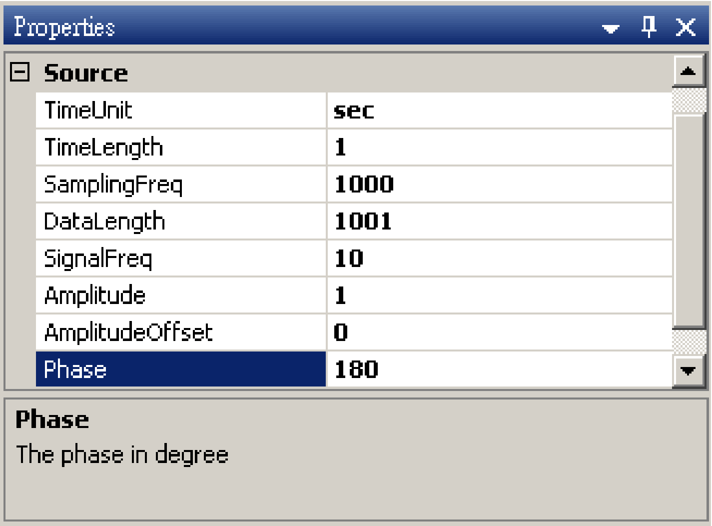
Create a since wave in Netowrk panel using *Source / Sine Wave* and its default frequency is10Hz. Then add two more sine waves. One is set with *Properties / SignalFreq* = 5Hz and the other is set with *Properties / SignalFreq* = 10Hz *and Properties / Phase* = 180 degree (unit is in Degree). Finally, merge three signals into a Multi-channel signal using *Conversion / Merge to Multi-channel*. In this case, three sine waves are generated: 10Hz, 5Hz, and 10Hz with phase shift of 180 degree. Using *Viewer / Channel Viewer*, the signals can be displayed. Black curve is Sine, blue curve is Sine2, and red curve is Sine3.

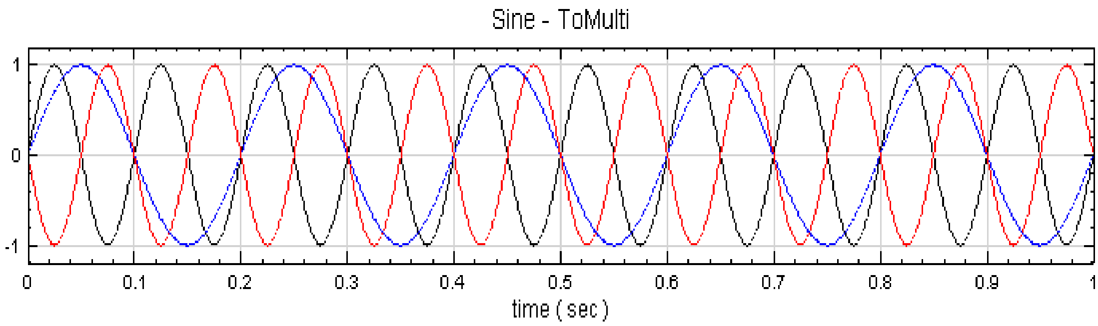


Sine2 Properties



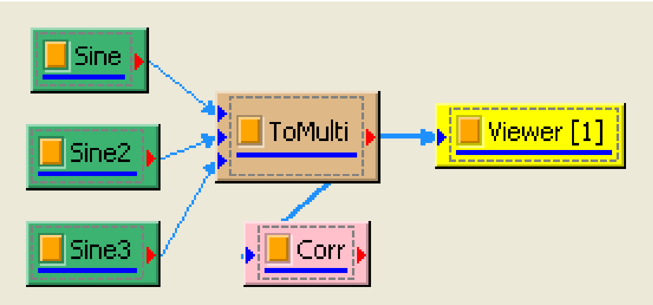
Sine3 Properties

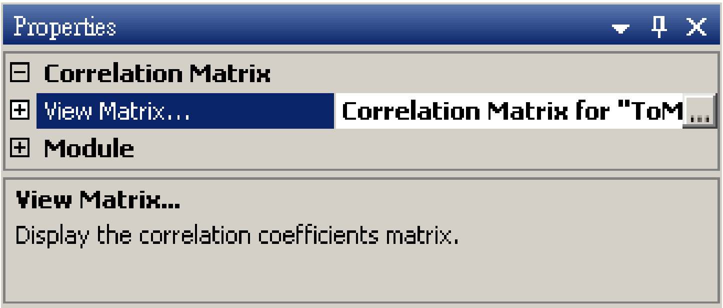


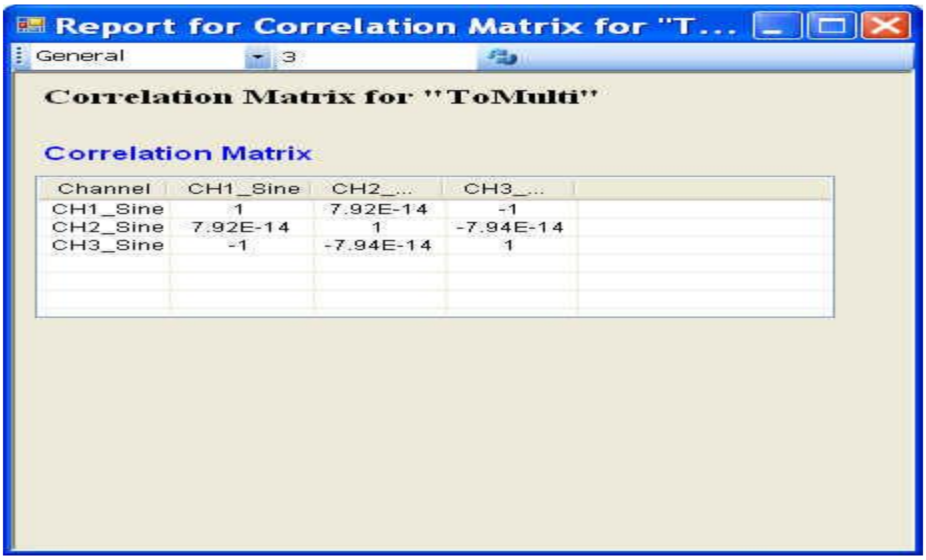


Connect ToMulti SFO to *Compute / Statistics / Correlation Matrix* and click *Properties / View Matrix…* to show the result in the pop-up window. The matrix element R_{ii} along the diagonal is self correlation value which is 1. R_{12} , R_{21} are correlation between Sine and Sine2. Since the value is very small, it means there is no correlation between these two signals. R_{32}, R_{23} are also very small. It means that there is no correlation between Sine2 and Sine3. Signals Sine and Since3 mirror

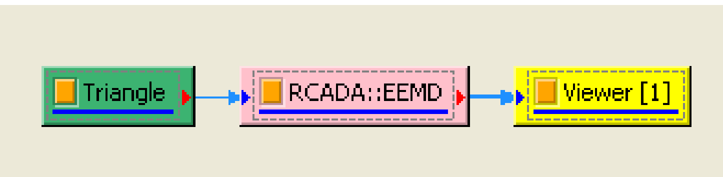
along X axis and the correlation is -1, it shows that two signals are totally negative correlated.

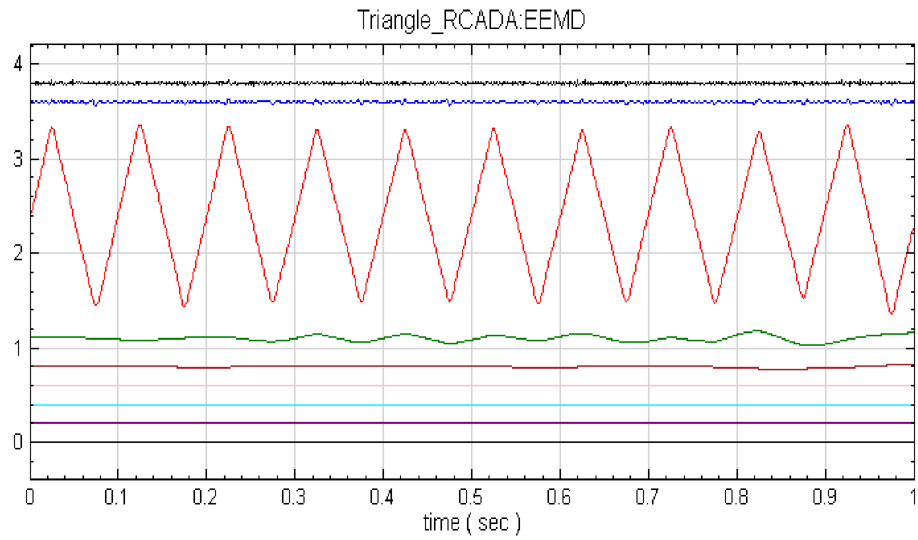




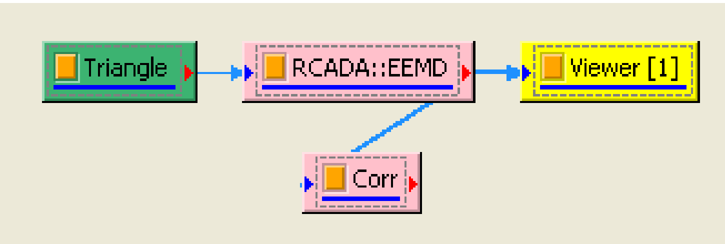


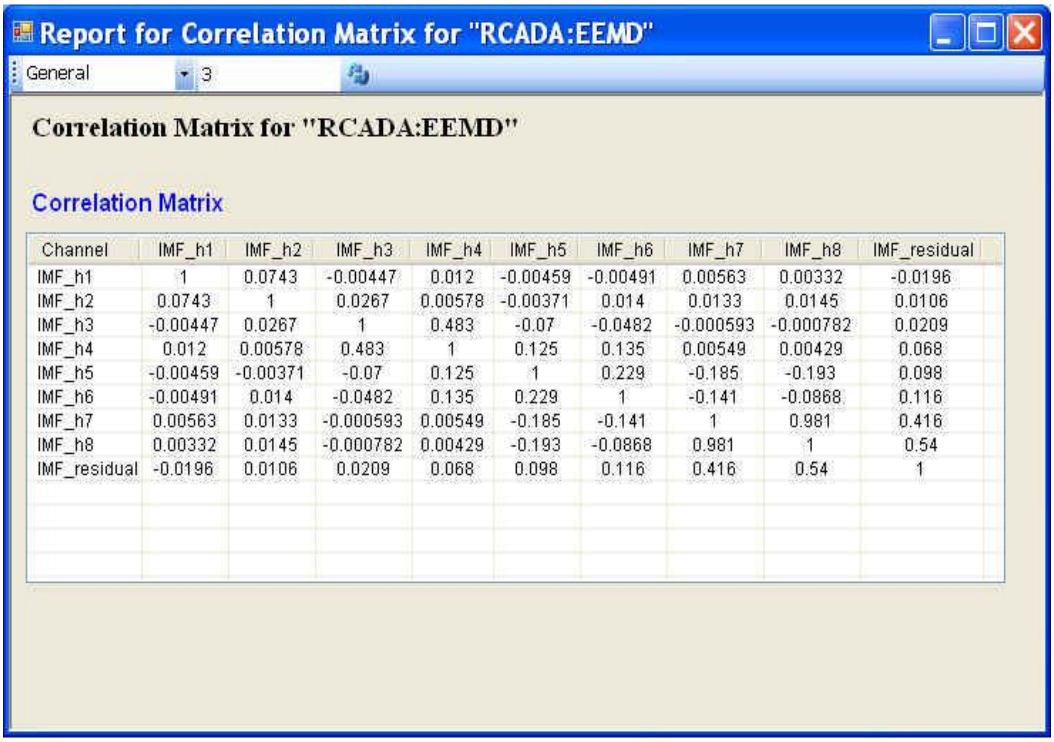
Create a new signal with *Source / Triangle Wave* and connec it to *Compute / HHT / RCADAEEMD* to calculate its IMFs. The results are displayed using *Channel Viewer.* Set *Properties / Multi-channel Display* in the view to List and set Viewer Height to 350. There are 9 channels in the signal.





Connect *RCADA EEMD* SFO to *Compute / Statistics / Correlation Matrix* for calculating correlation matrix among IMFs.




Report for Correlation Matrix for "RCADA:EEMD"

General 3

Correlation Matrix for "RCADA:EEMD"

Correlation Matrix

Channel	IMF_h1	IMF_h2	IMF_h3	IMF_h4	IMF_h5	IMF_h6	IMF_h7	IMF_h8	IMF_residual
IMF_h1	1	0.0743	-0.00447	0.012	-0.00459	-0.00491	0.00563	0.00332	-0.0196
IMF_h2	0.0743	1	0.0267	0.00578	-0.00371	0.014	0.0133	0.0145	0.0106
IMF_h3	-0.00447	0.0267	1	0.483	-0.07	-0.0482	-0.000593	-0.000782	0.0209
IMF_h4	0.012	0.00578	0.483	1	0.125	0.135	0.00549	0.00429	0.068
IMF_h5	-0.00459	-0.00371	-0.07	0.125	1	0.229	-0.185	-0.193	0.098
IMF_h6	-0.00491	0.014	-0.0482	0.135	0.229	1	-0.141	-0.0868	0.116
IMF_h7	0.00563	0.0133	-0.000593	0.00549	-0.185	-0.141	1	0.981	0.416
IMF_h8	0.00332	0.0145	-0.000782	0.00429	-0.193	-0.0868	0.981	1	0.54
IMF_residual	-0.0196	0.0106	0.0209	0.068	0.098	0.116	0.416	0.54	1

The diagonal of the matrix is self correlation, so the value is 1. For the other elements, the maximum value is between IMF_h7 and IMF-h8. It means that there exists strong correlation between them. Most correlation absolute values among elements are less then 0.1, it means that there is no correlation between them.

Related Functions

Covariance Matrix, Orthogonality Matrix, Merge To Multi-Channel, Channel Viewer, RCADA EMD.

Reference

Cohen, J., Cohen P., West, S.G.,&Aiken, L.S. (2003). Applied multiple regression/correlation analysis for the behavioral sciences. (3rd ed.) Hillsdale, NJ: Lawrence Erlbaum Associmtes.

3.4.4 Equiphase Statistics

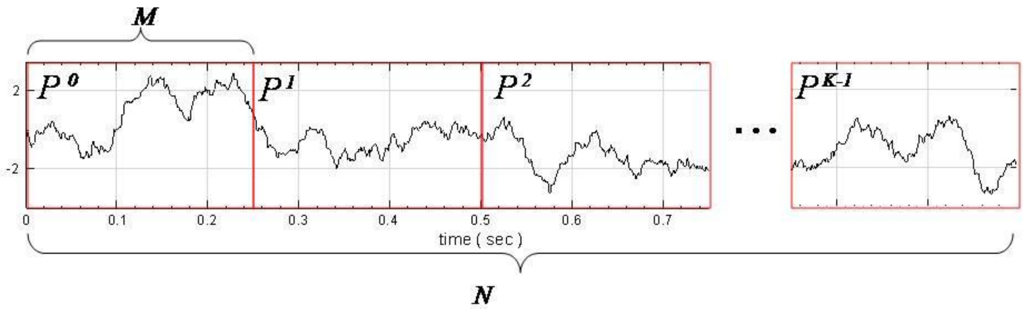
For a fixed period length M, Equiphase Statistics calculates the statistics of the same phase under this period, such as month average, week average etc.

Introduction

Let $X = \{x_0, x_1 .. x_{N-1}\}$ be a series, the length is N. The period of Equiphase Statistics is set to M, M<N. Then this series can be divided into K = Ceiling(N/M) small series. Ceiling(x) means to map the number to the smallest integer not less than x. Let small series be reprensented by P_j^k, where k is the small series number, j is the element in the series, then

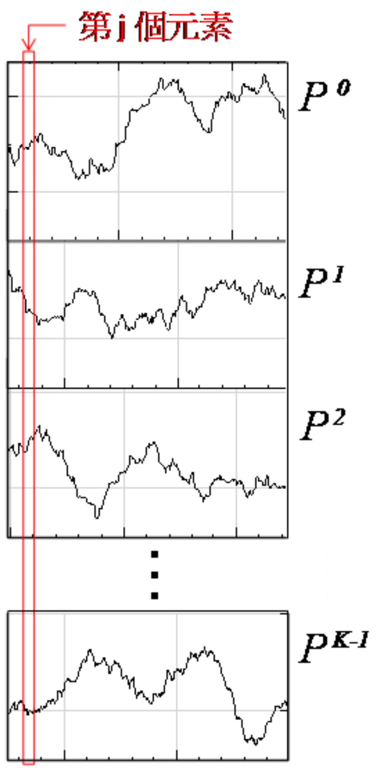
$$P_j^k = \{x_{j+k \cdot M}\},\ 0 \le j \le M-1,\ 0 \le k \le K-1$$

As shown below,



Equiphase Statistics picks elements with the same phame in each P_j^k as a group to calculate its statistical values. For example, equiphase mean is calculated as following:

$$EM_j = \frac{\sum_{k=0}^{K-1} P_j^k}{K},\ 0 \le j \le M-1$$

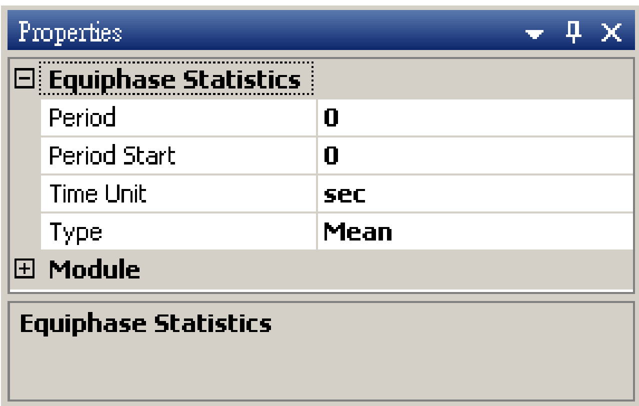


Please note that if N / M is not an integer, the length of the last small series M_last<N, so if the element j<=M_last, the number of calculated element is K; if the element j>M_last, the number of calculated element is K-1.

Equiphase statistics can calculated many different statistical values, including the values in basic statistics. Most of them are the same as the ones described in Basic statistics module. Some of them, such as First quartile, Third quartile, and Quantile, are described in Quartiles and Quantiles module.

Properties

This module accepts input of real number, single channel, and multi-channel signal. Paremeters are defined as following,



Property Name	Property Definition	Default
Period	Set Period time	10% of the total time length of the input signal
Period Start	Set Period Start time	0
TimeUnit	Set Unit of Time	sec
Type	Set statistics to be calculated	Mean

The option of Type is listed below. These statistics is calculated within the window.

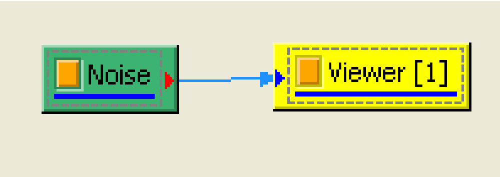
Type Options	Option Definition
Sum	Sum of the series
Min	Minimum
Max	Maximum
Mean	Mean value
Geometric Mean	Geometirc Mean Value
Harmonic Mean	Harmonic Mean Value
Trimmed Mean	Trimmed Mean
First quartile	1^{st} 4-quantile
Median	Median value
Third quartile	3^{rd} 4-quantile
Quantile	Quantile Number
StdDev	Standard Deviation
Variance	Variance
VarianceCoef	Variance Coefficient
Skewness	Skewness Value
Kurtosis	Kurtosis Value
SemiVariance	SemiVariance
SemiStdDev	SemiStdDev

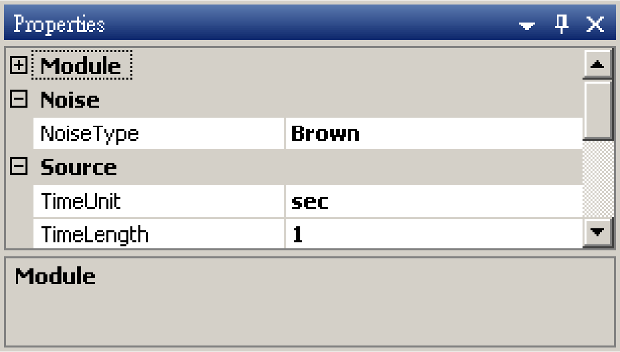
Some of the options may need to set parameters. Quantile is explained in Quartiles and Quantiles section. The rest of the options is detailed in Basic Statistics section.

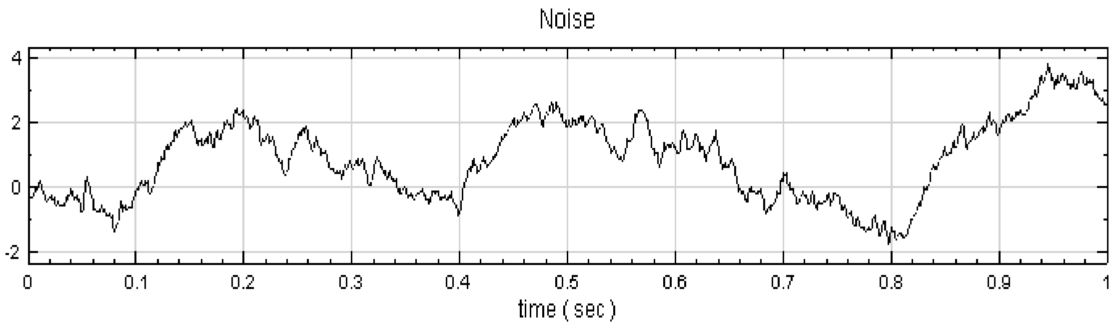
Example

Calculate different Equiphase Statistics values using a Brownian Noise.

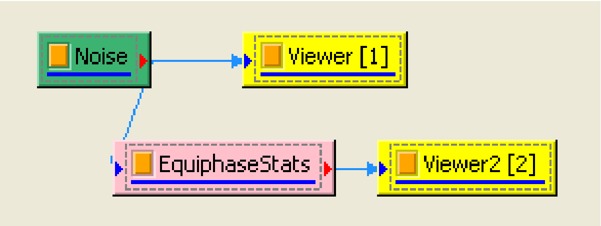
Create a signal by right clicking *Source / Noise* in Network panel and set *Properties / Noise Type* to Brown. Display the signal using *Viewer / Channel Viewer*.

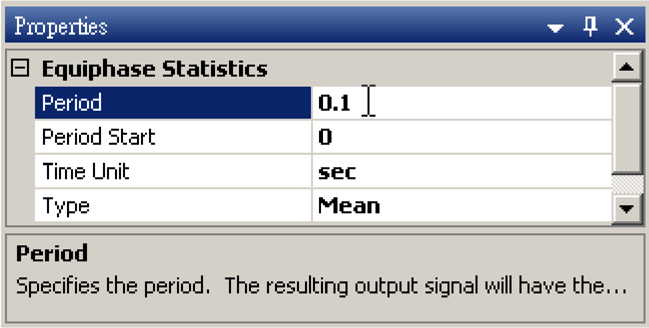


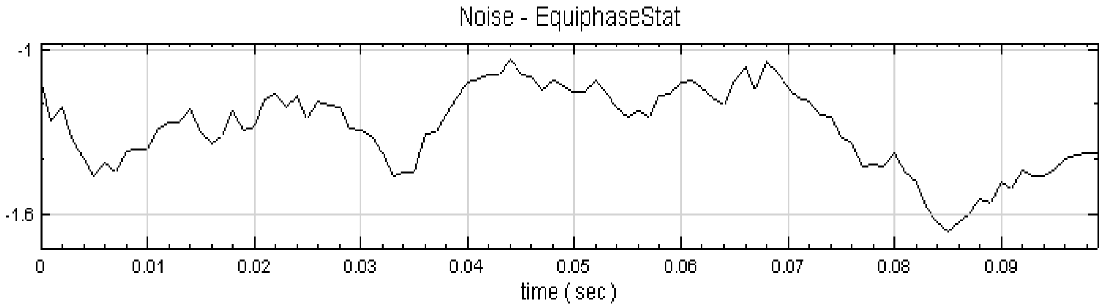




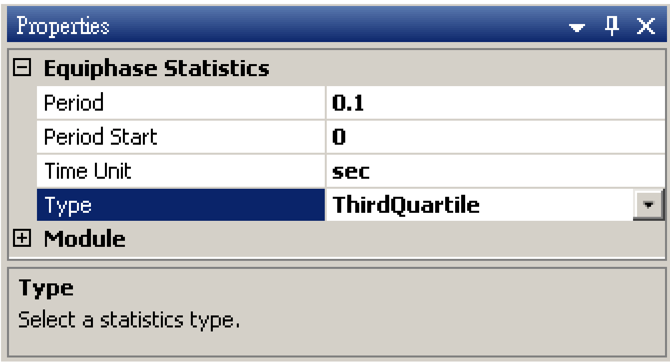
Connect Noise SFO to *Compute / Statistics / Equiphase Statistics*, the default type is Mean. Set *Properties / Period* to 0.1. It means the period is 0.1 second. The mean value of each element within the period is calculated. Display the result using *Viewer / Channel Viewer*.

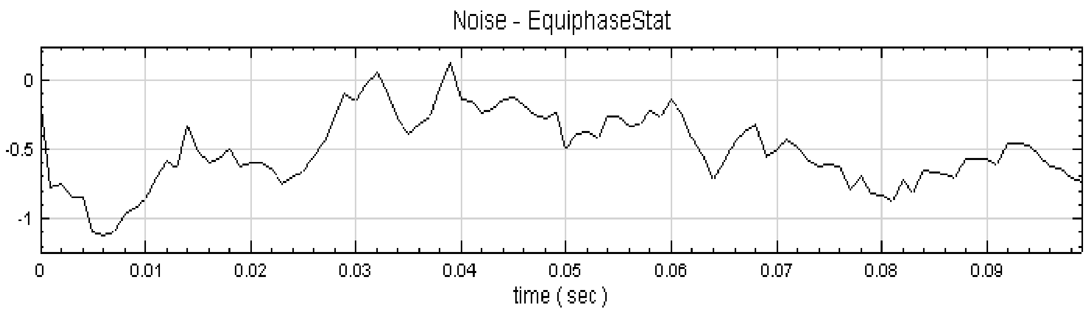






Change Type to Third Quartile. The elements of the 3rd quartile position in the period are calculated.





Related Functions

Basic Statistics, Rolling Statistics, Quartiles and Quantiles, Channel Viewer.

Reference

1. Michel Loeve, "Probability Theory", Graduate Texts in Mathematics, Volume 45, 4th edition, Springer-Verlaf, 1977

2. Joanes, D. N.&Gill, C. A. (1998) Comparing measures on sampleskewness and kurtosis. Journal of the Royal Statistical Society (Series D): The Statistician 47 (1), 183–189.

3.4.5 Kernel Smooth Density

Kernel smoothing density estimation is to calculate probability density function using non-parametric method.

Introduction

Let $X = \{x_0, x_1 .. x_{N-1}\}$ be a series, its kernel density estimation is:

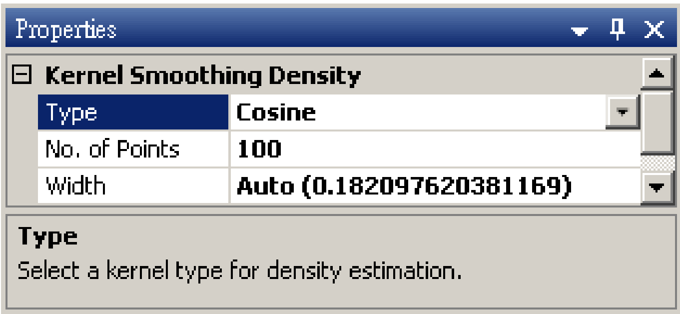
$$\hat{F} = \frac{1}{N}\frac{1}{h}\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$$

where h controls the smoothness, i.e. the size of the smoothing window, K is kernel function. This method applys kernel function to every decrete point and superposite the results of each point for smoothing the series. The concept is similar to histogram.

Properties

This module accepts input signals of real number, single channel or multi-channel, regular; the formats for output signals are real number, multi-channel, and regular.

The definitions for properties are listed below. Please note that for the output signal format for KS Density, one group of input signal may generate a group of two-channel output signals. The first channel is the X-axis, value ranging between the minimum and maximum of the input values; the second channel is the Y-axis, value being the corresponding probalility density value of the series value.



Properties	Property Definitions	Default value
Type	The types of kernel function, build-in function include: Uniform, Triangle, Epanechnikov, Quartic, Triweight, Gaussian, and Cosine. Their definitions are in the next table.	Gaussian
No.of Points	Descret points of the output signal.	100

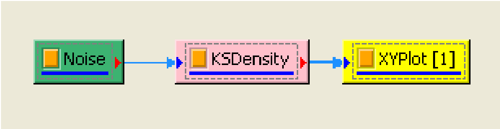
Width	The width of the sliding window, h, is the constant to control the smoothness level. The default value “Auto” is the best width calculated under the standard normal distribution.	Auto

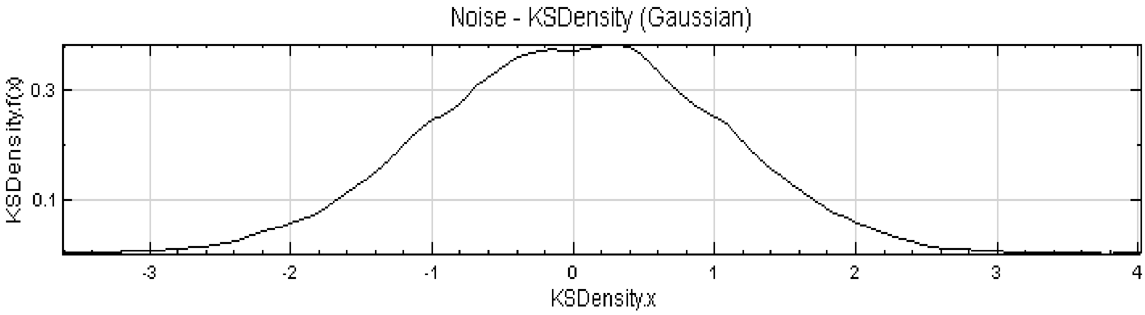
The following table lists common definitions of the kernel functions:

Type of Kernel function	**Definition**
Uniform	$K(u)=\begin{cases}\frac{1}{2}, & \lvert u\rvert\le 1\\ 0, & otherwise\end{cases}$
Triangle	$K(u)=\begin{cases}1-\lvert u\rvert, & \lvert u\rvert\le 1\\ 0, & otherwise\end{cases}$
Epanechnikov	$K(u)=\begin{cases}\frac{3}{4}(1-u^2), & \lvert u\rvert\le 1\\ 0, & otherwise\end{cases}$
Quartic	$K(u)=\begin{cases}\frac{15}{16}(1-u^2)^2, & \lvert u\rvert\le 1\\ 0, & otherwise\end{cases}$
Triweight	$K(u)=\begin{cases}\frac{35}{32}(1-u^2)^3, & \lvert u\rvert\le 1\\ 0, & otherwise\end{cases}$
Gaussian	$K(u)=\begin{cases}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}, & \lvert u\rvert\le 1\\ 0, & otherwise\end{cases}$
Cosine	$K(u)=\begin{cases}\frac{\pi}{4}cos(\frac{\pi}{2}u), & \lvert u\rvert\le 1\\ 0, & otherwise\end{cases}$

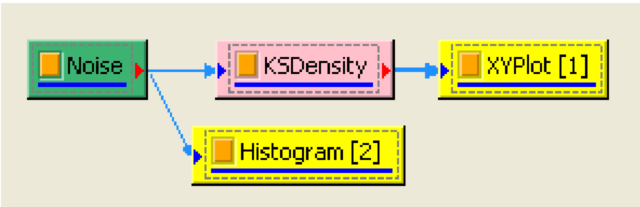
Examples:

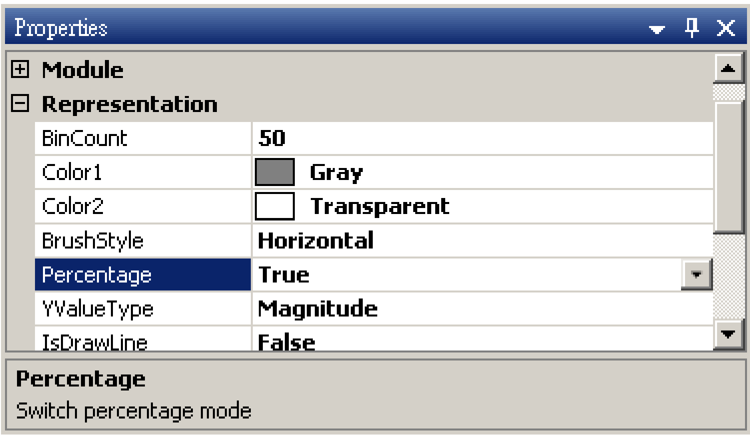
Creat a signal using Source / Noise, adjust its Properties / Noise Type to Gaussian, Time Length to 10 seconds, connect to Kernel Smooth Density, then link to XY Plot.

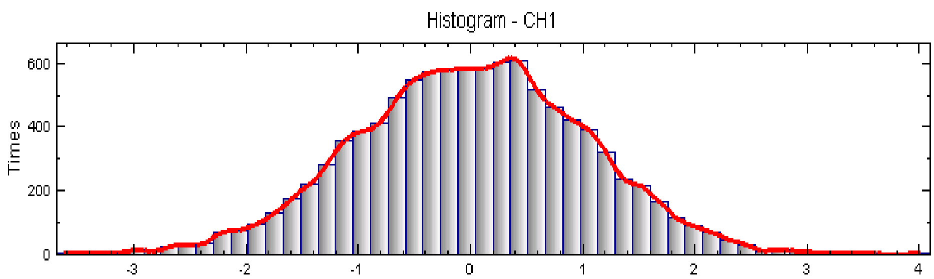




On the other hand, directly gragh Noise using the Viewer / Histogram Viewer, set Properties / BinCount of the histogram to 50, Percentage to True, then the following histogram is shown in the viewer.

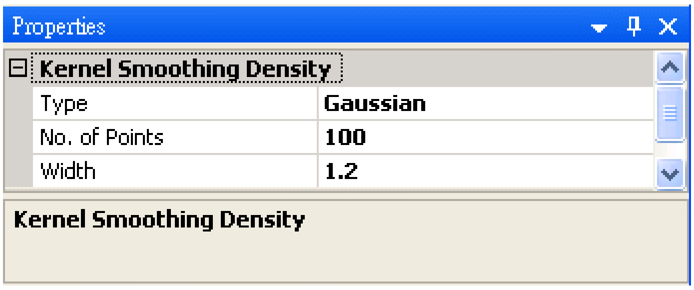


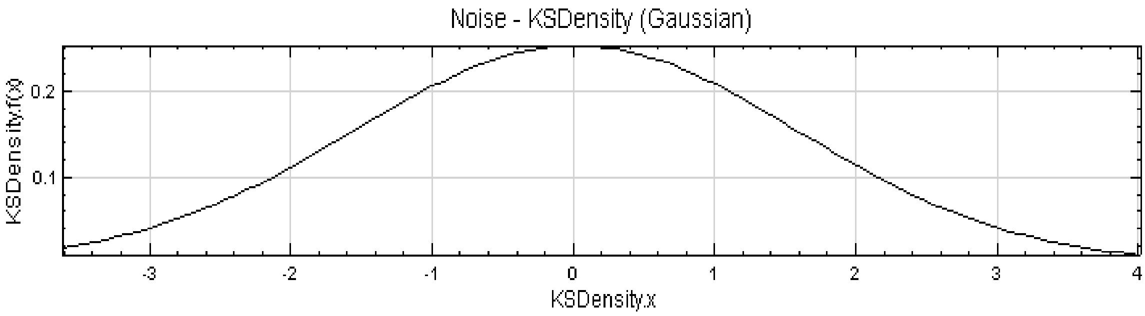




The basic concept of the two graphs is the same. Kernel smooth density uses kernel function to express probability as a continuous density function, while the histogram uses the value interval to calculate the probability of occurrences, and in this model the area above X-axis is 1 for the density function, whearas the Y-axis of the histogram is directly the occurrences.

If you adjust the Properties / Width of KSDensity a little bigger to 1.2, you can see that the resulting curve is smoother.





Related Functions

Histogram, Noise, XY Plot.

Reference

T. Hastie, R. Tibshirani and J. Friedman. The Elements of Stbtistical Learning, Chapter 6, Springer, 2001.

3.4.6 Orthogonality Matrix

Orthogonal matrix is calculation of dot product of the serie. If the two signals are orthogonal, the value will be zero. It can be used to determine whether the IMFs calculated from EMD are orthogonal.

Introduction

Let $X=\{x_0, x_1..x_{N-1}\}$, $Y=\{y_0, y_1, ..., y_{N-1}\}$ are two series. The orthogonality is defined as the inner product of the two series, calculated as follows:

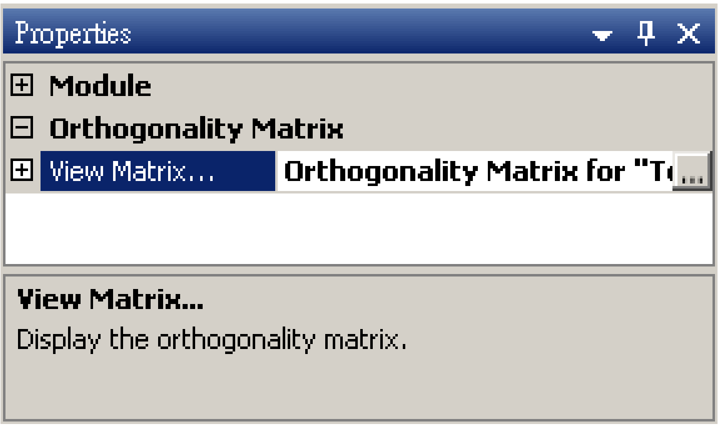
$$orth_{xy}=\frac{X}{\|X\|}\cdot\frac{Y}{\|Y\|}=\frac{\sum_{i=0}^{N-1} x_i y_i}{\sqrt{\sum_{i=0}^{N-1} x_i^2 \cdot \sum_{i=0}^{N-1} y_i^2}}$$

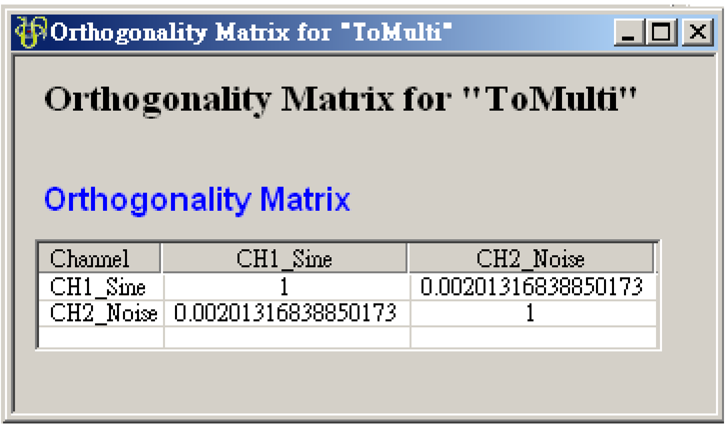
If there are M series, their corresponding orthogonal matrix as follows:

$[O_{lk}]=orth_{lk}$, Where l and k are for the channel number

Properties

This module accepts the input of real number, multi-channel, regular signals. The output is a M x M square matrix, M for the number of channels of the input signal. The output format is the indexed data values. In the Properties / View Matrix ... you can use the Reporter window to see the results.



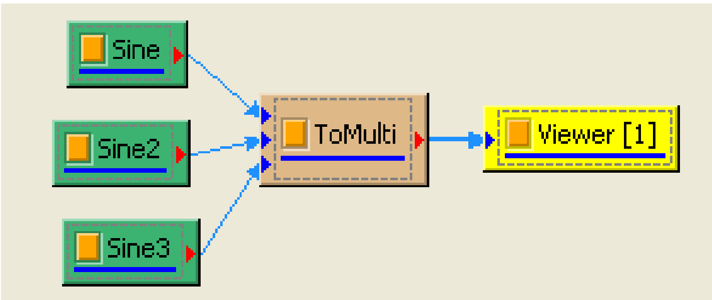


Channel	CH1_Sine	CH2_Noise
CH1_Sine	1	0.00201316838850173
CH2_Noise	0.00201316838850173	1

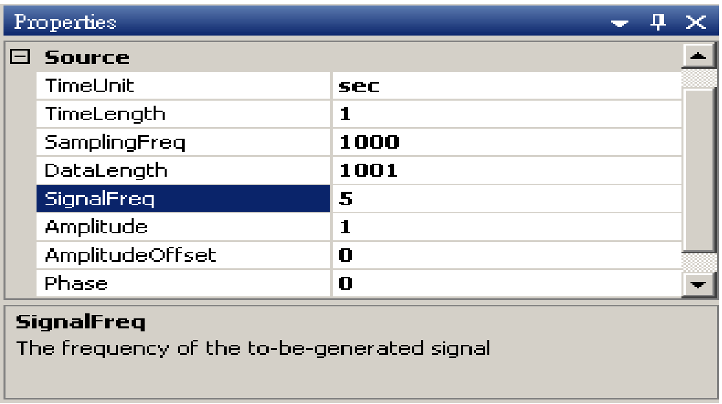
Example

Different phase angle and frequency of sine wave as the input signal to calculate orthogonal matrix:

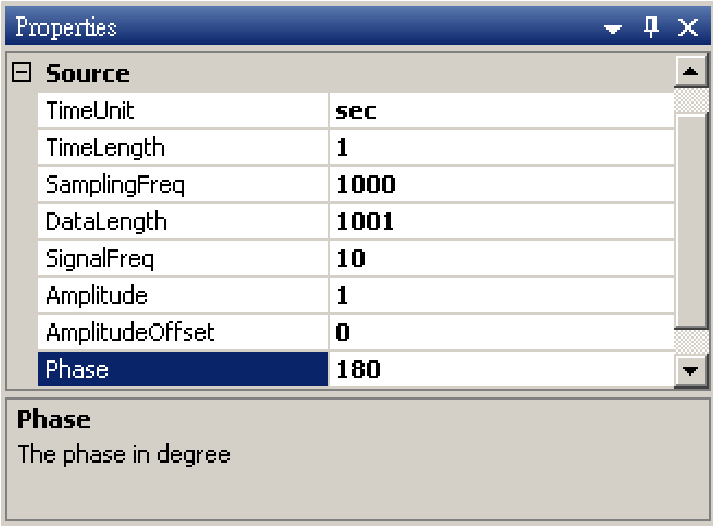
In the Network window, right press, select “Source / Sine Wave” creates a sine wave with default frequency of 10 Hz. Next create two sine waves. One wave set “properties/ SignalFreq = 5Hz” and the other wave set “Properties / Phase =180 degrees”. Lastly, use “Conversion / Merge to Multi-channel” to combine three waves into a Multi-Channel signal. The above steps create a sine wave with frequency of 10Hz, a sine wave with frequency of 5Hz and a sine wave with 180 degree phase angel. Use “Viewer/Channel Viewer” to graph result, with the black line representing the Sine, the Blue line representing Sine2, and the red line on behalf of Sine3.

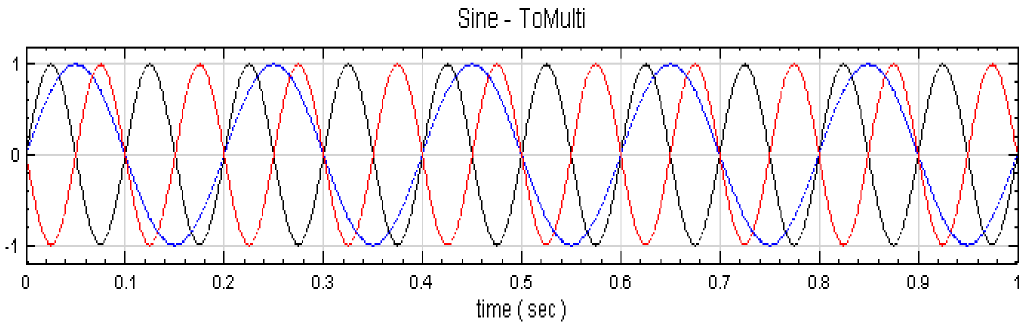


Sine2 Properties

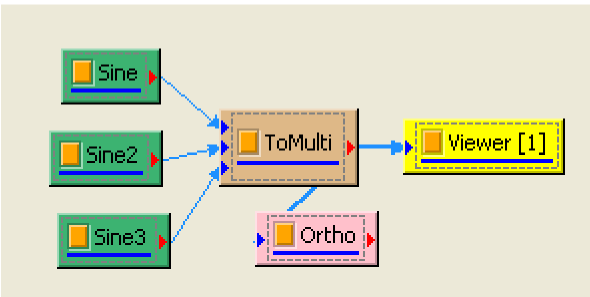


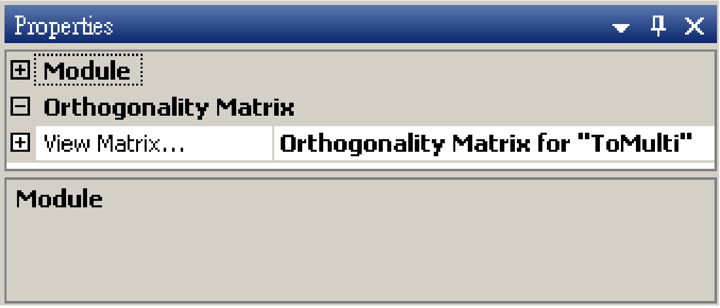
Sine3 Properties

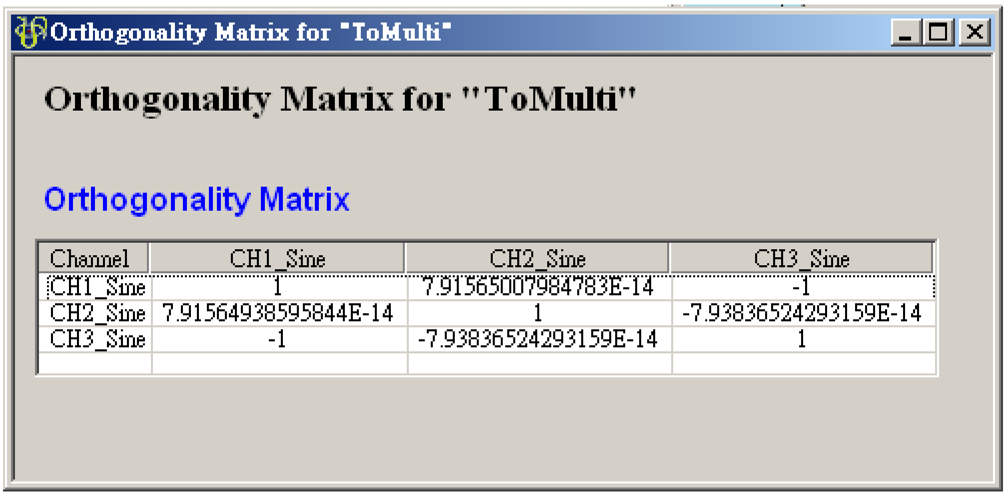




Connect "Compute / Statistics / Orthogonality Matrix" after "ToMulti", select "Properties / View Matrix" to show calculating results. The diagonal entry of the matrix O_{ii} is the inner product of its own signals. The value is 1; O_{12} , O_{21} are the comparision between Sine and Sine2. If the value is extremely small, meaning that the two signals are orthogonal. If O_{32}、 O_{23} are also extremely small, then Sine2 and Sine3 are orthogonal. Sine and Sine3 are series symmetrical to the X axis, so its result is the -1.







Channel	CH1_Sine	CH2_Sine	CH3_Sine
CH1_Sine	1	7.91565007984783E-14	-1
CH2_Sine	7.91564938595844E-14	1	-7.93836524293159E-14
CH3_Sine	-1	-7.93836524293159E-14	1

Related Function

Covariance Matrix, Correlation Matrix, Merge To Multi-Channel, Channel Viewer.

Reference

Probability, Random Variables and Stochastic Processes. McGraw-Hill, Page 211.

3.4.7 Quartiles and Quantiles

Quantile is the element value at certain percentage position of a sorted series, while the quartile is the element value at 25% position, 50% position, and 75% position of the sorted series.

Introduction

Let $X = \{x_0, x_1..x_{N-1}\}$ be a serie with N elements, the quartile can be expressed as:

$$P(X \le x_q) \le p = 1/4$$

More specifically, the quartile of a series is the value at cumulative distribution function equal to 25%, position q at N * 25% - 1. The median and three quarters of the median follows the same concept. Quantile is more generalized and use percentage as standard. For example 17th quantile represents values with the cumulative distribution function equals to 17%.

If the location of quantile is between 2 points (q = N * p - 1 is not an integer), we need to estimate the location. The estimation methods are plenty. This module presents 5 most common used methods for the users to choose:

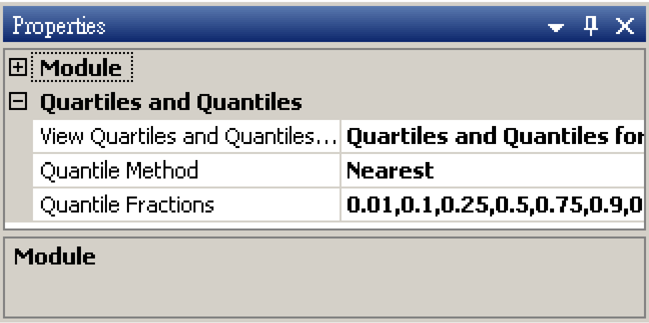
Interpolation	**Definition**	**Notes**
Linear	$x_q = x_i + (q-i)(x_{i+1} - x_i)$ Which i represents the integer part of q	The values before and after the quartile position are used for the linear interpolation quantile.
Next point	$\begin{cases} x_{floor(q)}, & r=0 \\ x_{floor(q+1)}, & r>0 \end{cases}$	The value of the next position after the quantile position.
Average	$x_q = \frac{1}{2}(x_{i+1} + x_i)$	The average values of the position before and after the quantile position.
Weighted Average	$x_q = x_{i+1} + g(x_{i+2} - x_{i+1})$ Which i represents the integer part of (N -1) * p, and g is the fractional part.	The point before the quantile position plus the difference between the points before and after the quartile points. This is adopted by Microsoft Office Excel.

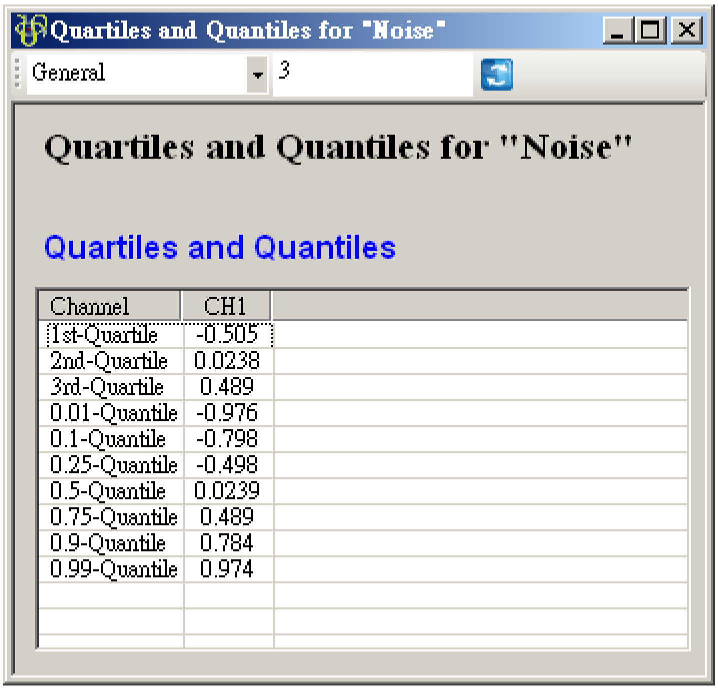
Nearest	$\begin{cases} x_i, & g<0.5 \\ x_{i+1}, & g\geq 0.5 \end{cases}$ Which i represents the integer part of (N-1) * p, and g is the fractional part.	The value of the position closest to the quantile position.

Properties

This module accepts input signals in real number, complex number, single channel or multi-channel and regular.

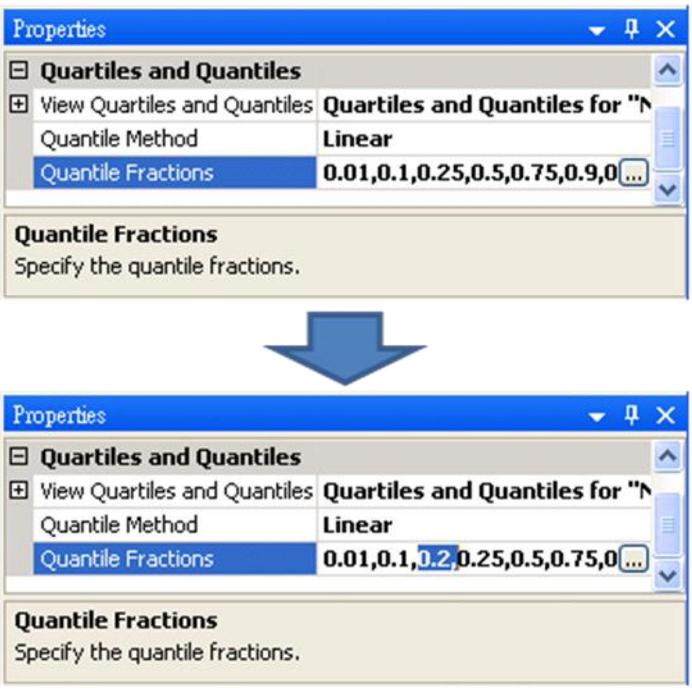
The interface for parameter setting is shown below. This module by default calculates quartile values of serie such as quartile, median, three-quarters of the median values. Also available under “Quantile Fractions” is quartile values to calculated next. “View Quartiles and Quantiles” will show pop-up window for the results. “Quantile Method” allows selection of the estimation methods for quantile. Here we describe the breakdown of various parameters below.



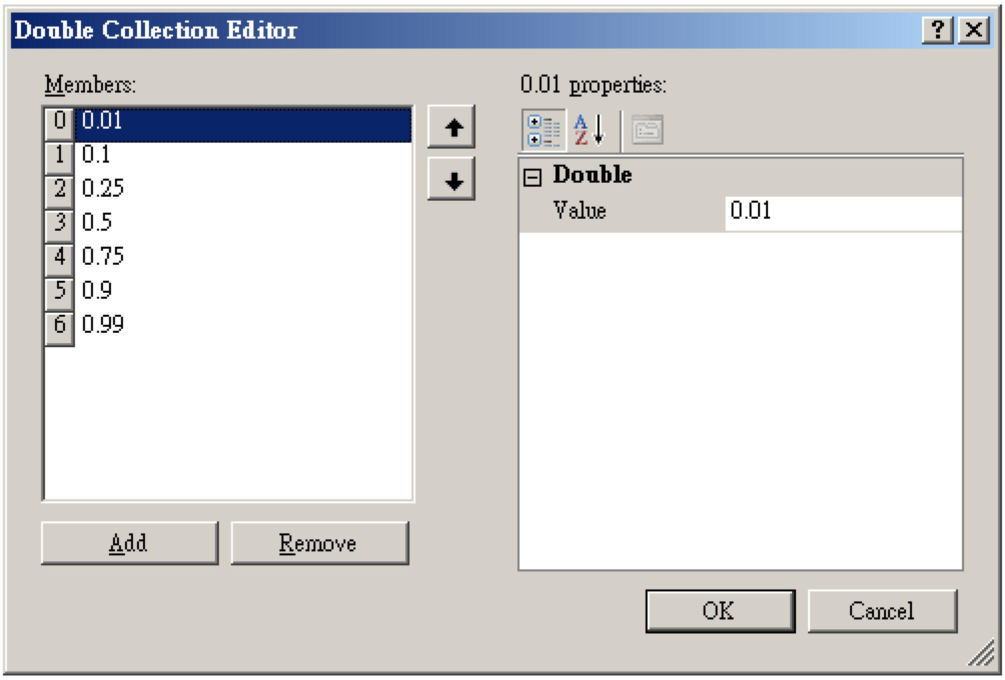


The above is the pop-up window when clicking button to the right of “View Quartiles and Quantiles”. The results of each calculation of the signal are presented in the columns with the first column showing the names of Quartile and Quantile. From the Second Column, each Column corresponds to the channel of each input signal. The first three rows calculate three quartile values, followed by rows calculating quantiles. Users can set parameters with “Quartile Fractions”. The following describes parameters for “Quantile Fractions”.

There are two ways to set quantiles fractions. The first method is directly to change the data in the fields, eg. between 0.1 and 0.25, type "0.2,” to add (see below).



The second method is to click button to the right of Quantile Fractions to pop up the quantile edit window (below). If you need to add or remove a number of quantile values, you can take this approach. On the left is the “Quantile Member” panel. User can use the “Add / Remove” button to add / remove members. Additionly, in the right panel you can edit each member’s quantile ratio. For example, 0.01 Representative 1% quantile, when the editing is complete, press the OK button to complete the setup. After setting up these parameters, the result of the Quantiles calculations will appear in “View Quartiles and Quantiles” window.



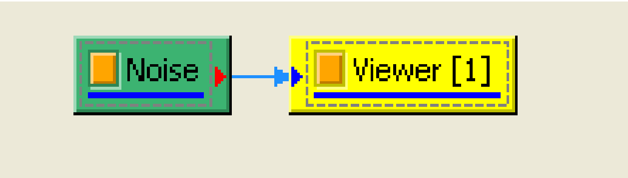
There are five Quantile Methods, namely linear, next, mean, weighted mean, and nearest. Their methods of estimation are distailed in the Introduction section.

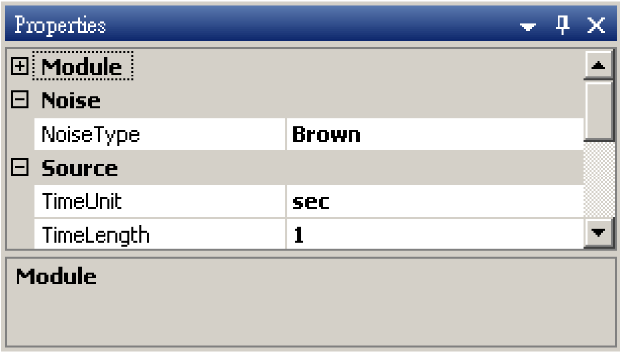
Parameter name	**Parameter Definition**	**Default**
View Quartiles and Quantiles	View results for Quartiles and Quantiles	n/a
Quantile Method	Linear, Next, Mean, Weighted mean, Nearest.	Linear
Quantile Fractions	Can set the percentage of multiple Quantiles	[0.01; 0.1; 0.25; 0.5; 0.75; 0.9; 0.99]

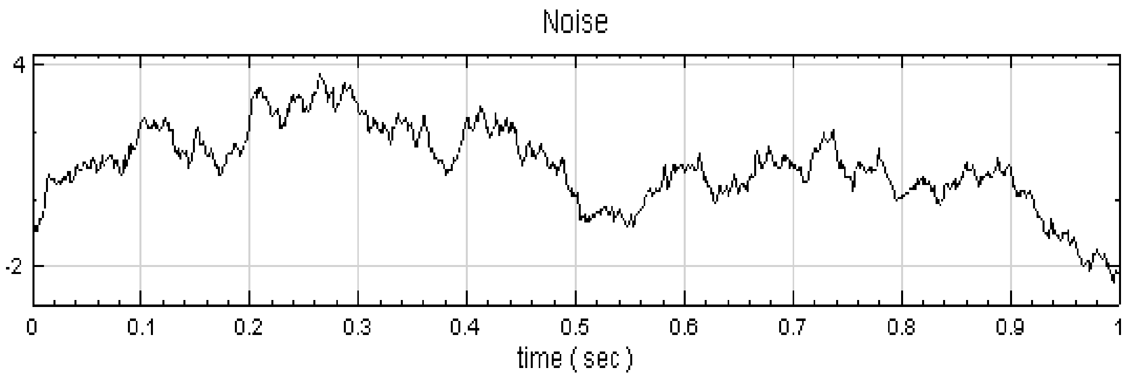
Examples

Use one group of Brownian Noise as input signal, calculate Quartiles and Quantiles.

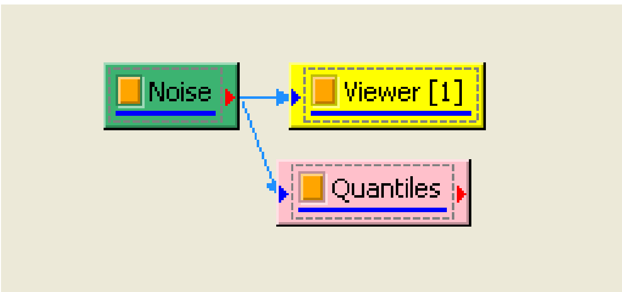
In Network panel, right click to add “Source/Noise”, adjust “Properties/Noise Type” to Brown, use “Viewer/Channel Viewer” to graph results.

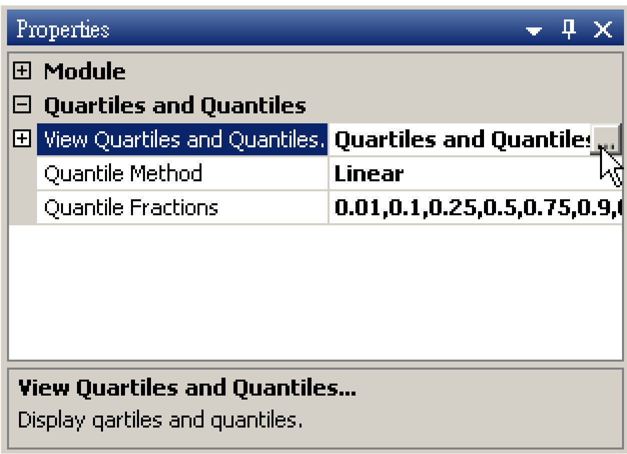


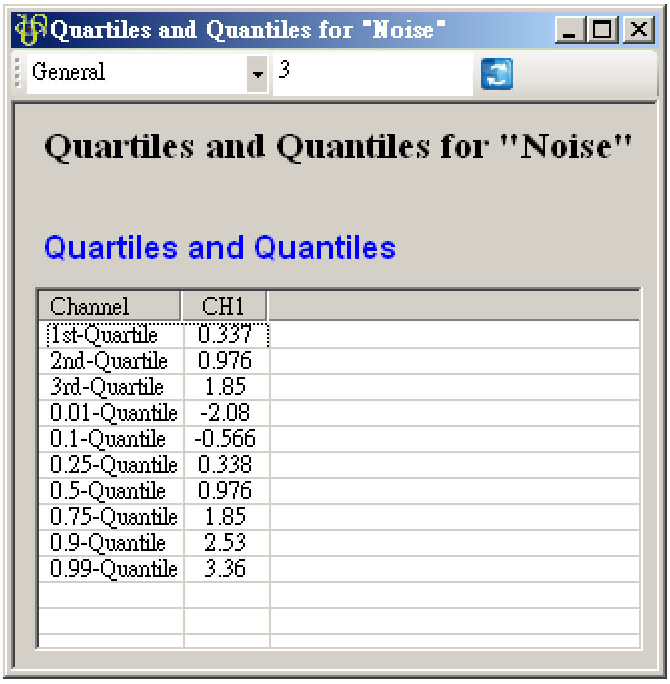




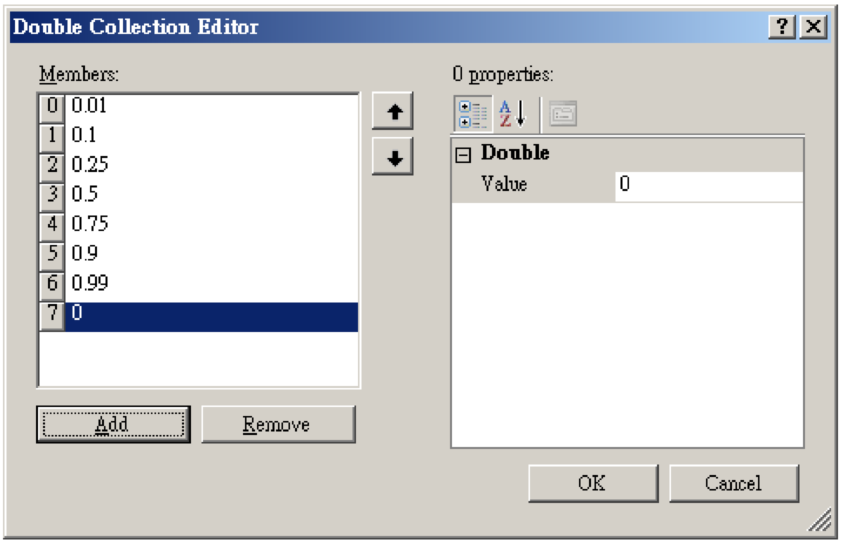
Connect “Compute/Statistics/Quartile and Quantiles” after “Noise” to calculate quartile values, click “Properties/View Quartiles andQuantiles” to examine results.



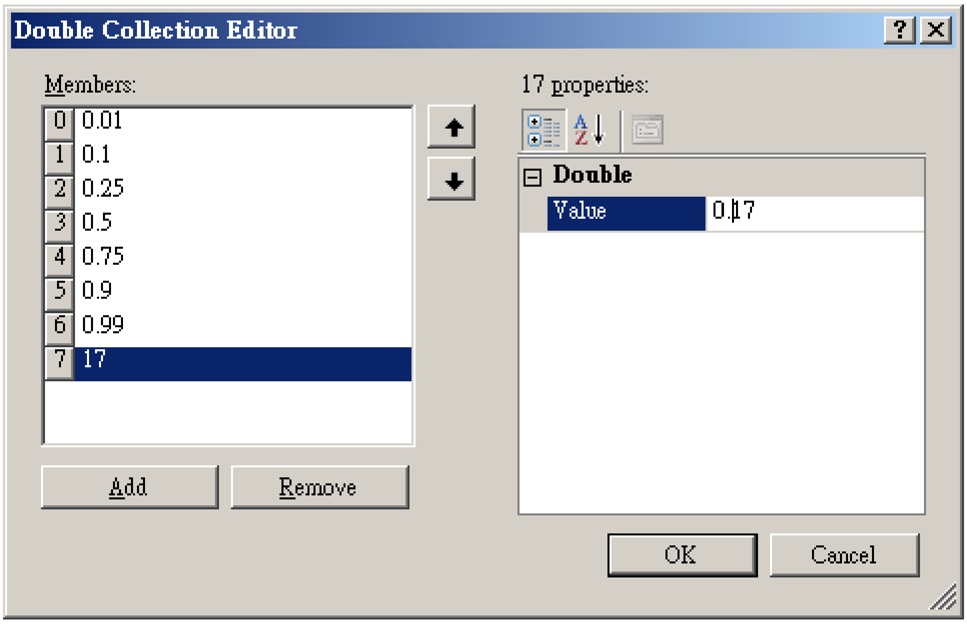




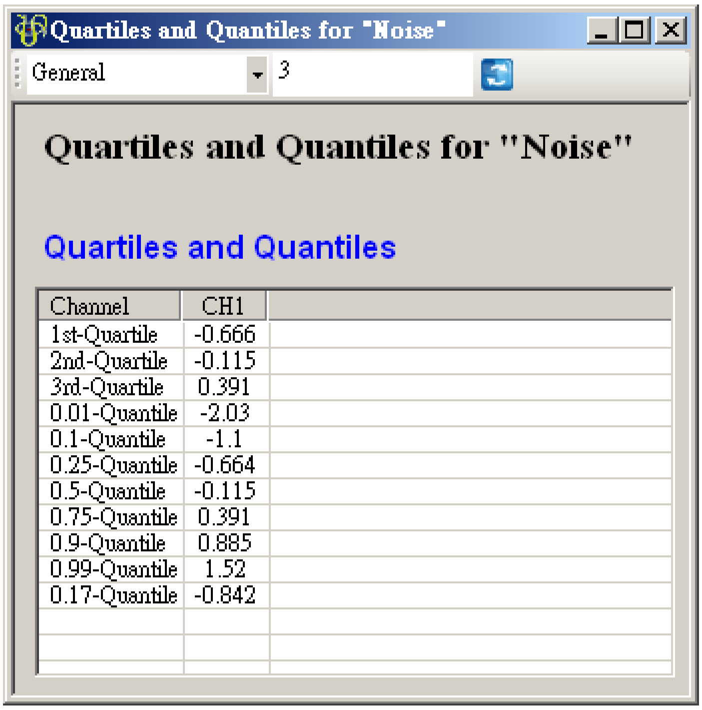
3. Select “Properties/Quantile Fractions” to edit the different quantile values. Press the “Add” button in the interface (below) will add a “0” under Members.



Then, on the righthand of the editor interface you can setup the percentage for the quantile. For example, set to 0.17, and then press OK.



4. Select “View Quartiles and Quantiles” again, you can see that the 17th quantile has been added.



Channel	CH1
1st-Quartile	-0.666
2nd-Quartile	-0.115
3rd-Quartile	0.391
0.01-Quantile	-2.03
0.1-Quantile	-1.1
0.25-Quantile	-0.664
0.5-Quantile	-0.115
0.75-Quantile	0.391
0.9-Quantile	0.885
0.99-Quantile	1.52
0.17-Quantile	-0.842

Related Functions

Basic Statistics, Rolling Statistics, Channel Viewer.

3.4.8 Rolling Statistics

Rolling statistics: Setup a window with the width of M elements, use statistical function to calculate the statistical value within the window, such as average, and move this window along the data to calculate statistics within the new window. This calculation method is called Rolling statistics.

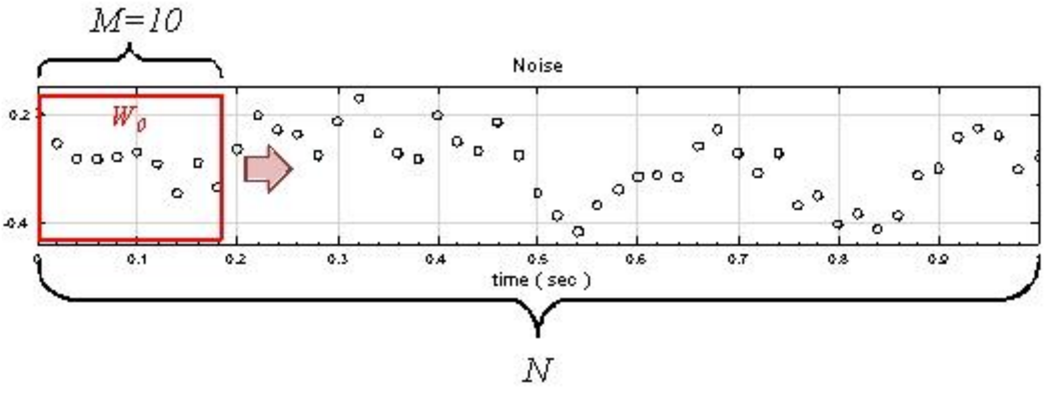
Introduction

Let $X = \{x_0, x_1..x_{N-1}\}$ be a data series with N members, and the width of the window W_j for the rolling statistics is M, M<N. The elements within the window can be expressed as $W_j = \{x_j, x_{j+1}, \ldots, x_{j+M-1}\}$ with $0 < j < N - M$. Rolling statistics is to calculate the statistical value within the window, such as Rolling mean.

$$\mu_i = \frac{\sum_{j=i}^{i+M-1} x_j}{M}, \quad 0 \le p < M$$

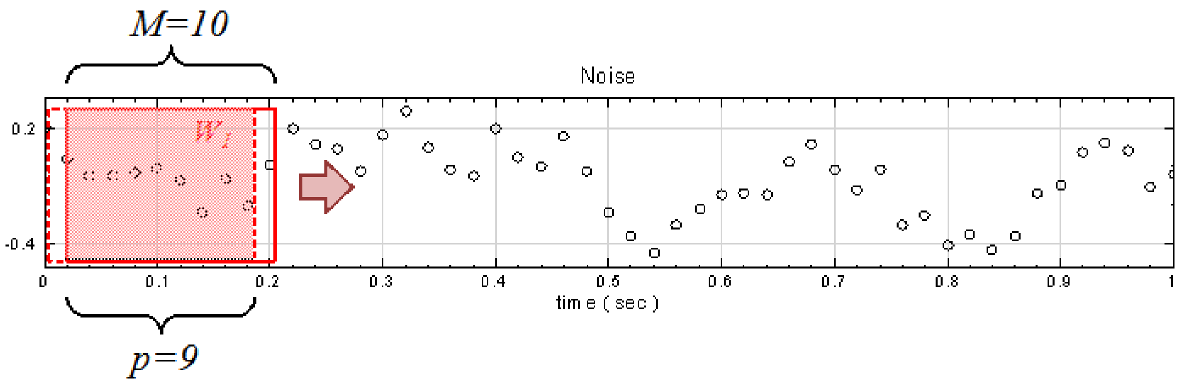
Next define an "overlap" value p, p represents the number of elements in the window overlapping with the previous one. Let use "rolling means" as an example. If the size of the window M = 10, then the 1st point in the output is

$$\mu_0 = \frac{\sum_{j=0}^{9} x_j}{10}$$, see the following diagram:



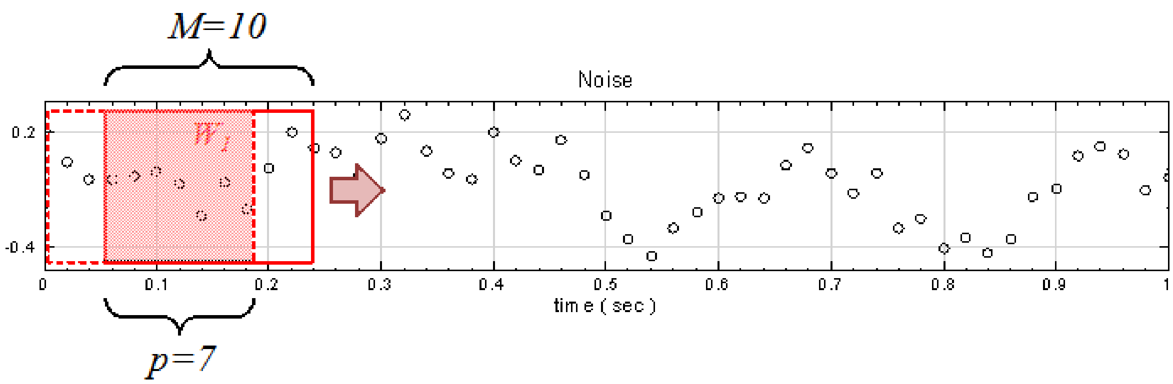
The ranges are signal position 0 to 9 as indicated above. If p = 9, then the ranges for the next window are signal position 1 to 10.

$$\mu_0 = \frac{\sum_{j=1}^{10} x_j}{10}$$



And there are 9 elements overlap between the 2 windows. If p = 7, then

$$\mu_0 = \frac{\sum_{j=3}^{12} x_j}{10}$$



The window has 7 position ovelap with the previous range. And so on, the output length of the series is:

$$K = \frac{N - p}{M - p}$$

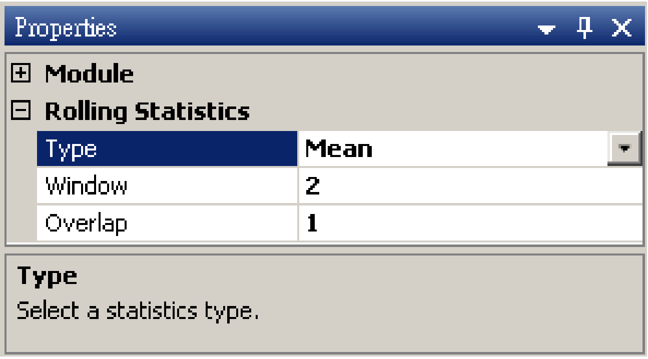
Please note that when “m - p>1”, the length of output series K may not be an integer. One solution is to round the remainder and keep the result for the complete window.

$$K = floor(\frac{N-p}{M-p})$$

Rolling statistics can be used to calculate statistical values similar to the functions in "Basic statistics" module and will not be repeated here.

Properties

This module accepts input signals in the formats of real number, complex number, single channel or multi-channel, and regular. The properties are defined as follows:



Properties	**Definition**	**Default**
Type	Options for statistical calculation, detailed list below	Mean
Window	Set the window size, the unit is the number of elements for the signal.	2
Overlap	Set the number of overlapping elements for the rolling window	Window width - 1

The options for "Type" are defined as follows; it will calculate the statistics within the window.

Options	**Definition**
Sum	Calculate sum
Min	The smallest number in the series
Max	The biggest number in the series
Mean	Calculate average
Geometric Mean	Calculate geometric mean
Harmonic Mean	Calculate harmonic mean

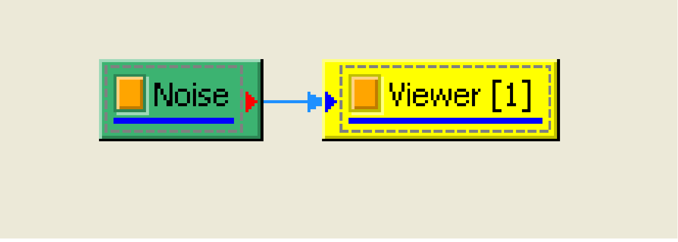
Trimmed Mean	The mean without the 1st and last numbers.
First quartile	1st quartile of the series
Median	The median of the series
Third quartile	The third quartile of the series
Quartile	Quartile of the series
StdDev	Calculate the standard deviation of the series
Variance	Calculate the variance of the series
VarianceCoef	Coefficient of variation
Skewness	The skewness of the series
Kurtosis	The kurtosis of the series
Semivariance	Semivariance of the series
SemiStdDev	Semi Standard deviation of the series

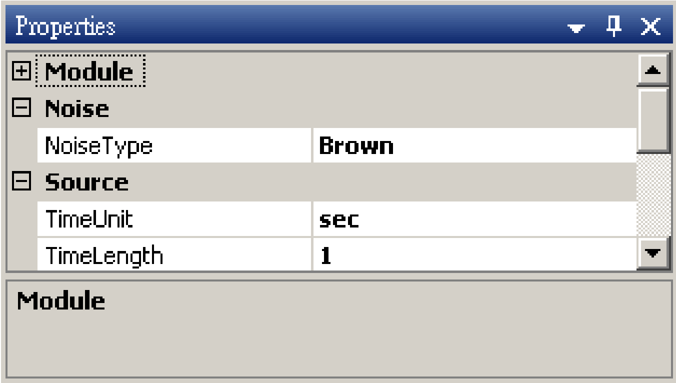
Some options have parameters that need to be set. The parameters for “Quantile” please refer to the documentation for “Quartiles and Quantiles”. For the definition of other statistics, please refer to the documention for “Basic Statistics”.

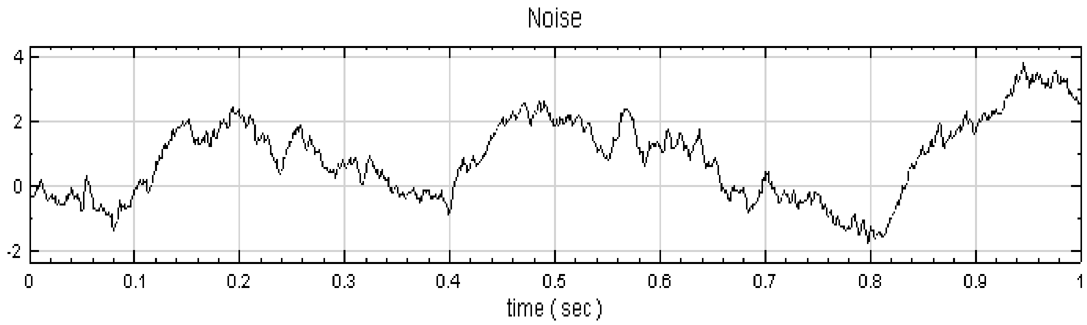
Examples

Use a set of Brownian Noise as input signal, calculate rolling statistics.

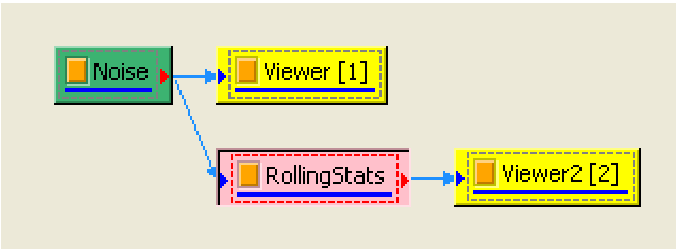
Right click in the Network panel to add “Source / Noise”, adjust “Properties/Noise Type” to Brown, then use “Viewer / Channel Viewer”to graph results.

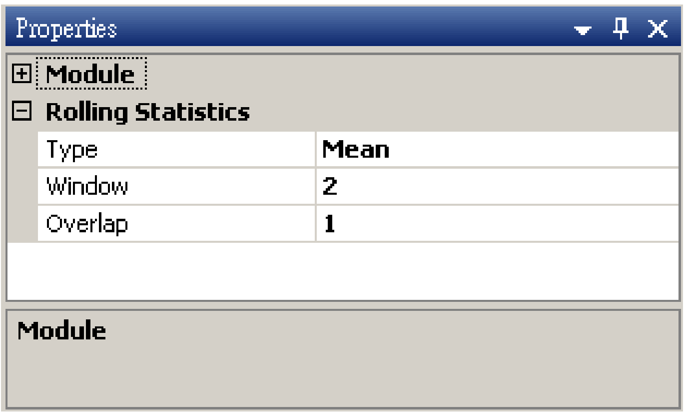


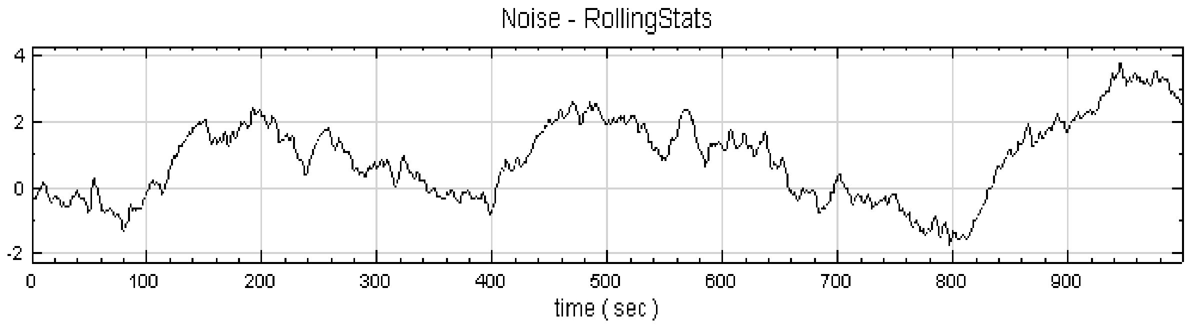




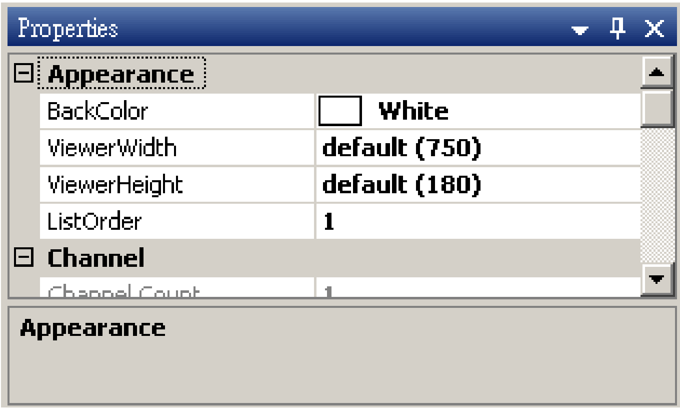
And connect “Compute /Statistics / Rolling Statistics” to the right of “Noise”. The Default statistics is Mean, window default width is 2. Graph results with Channel Viewer.

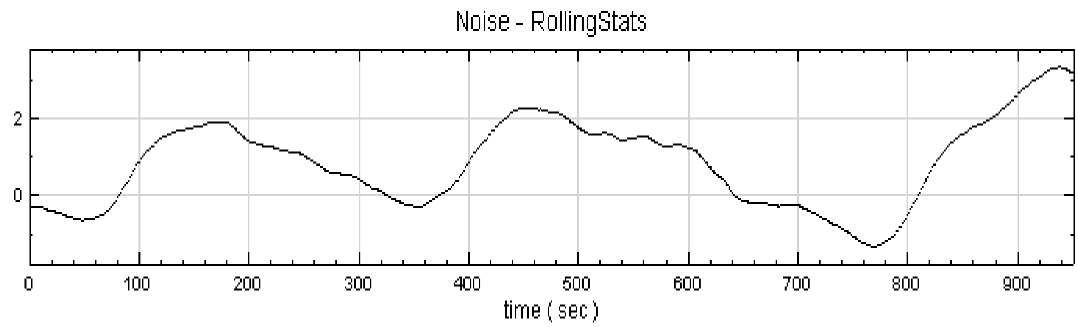






When window = 2, the results from step 2 is not much different from the original signal. Adjust window to 50, the result is graphed below. Then click “RollingStats” icon, press “Data viewer” on Network toolbar to see the length of Data count. The length is 952, this can be checked with K value described in the Introduction.





Related Functions

Basic Statistics, Equiphase Statistics, Quartiles and Quantiles, Merge To Multi-Channel, Channel Viewer.

3.4.9 Hypothesis Test

Make an appropriate temporary hypothesis about the population, and define a standard to reject the hypothesis based on random sampling distribution. If the sample data fall in the range of rejection, then reject the original hypothesis. Otherwise you must accept the hypothsis.

Introduction

Laboratory data often includes chance of errors, true differences or other influences. Hypothesis testing can be used to solve such problems. In general, it consists of three steps: setting hypothesis, selecting methods of testing and determining whether or not to accept the hypothesis.

Null Hypothesis: the differences among samples are purely from chance.

Alternative Hypothesis: sample differences are true diffences and influenced by some non-random cause.

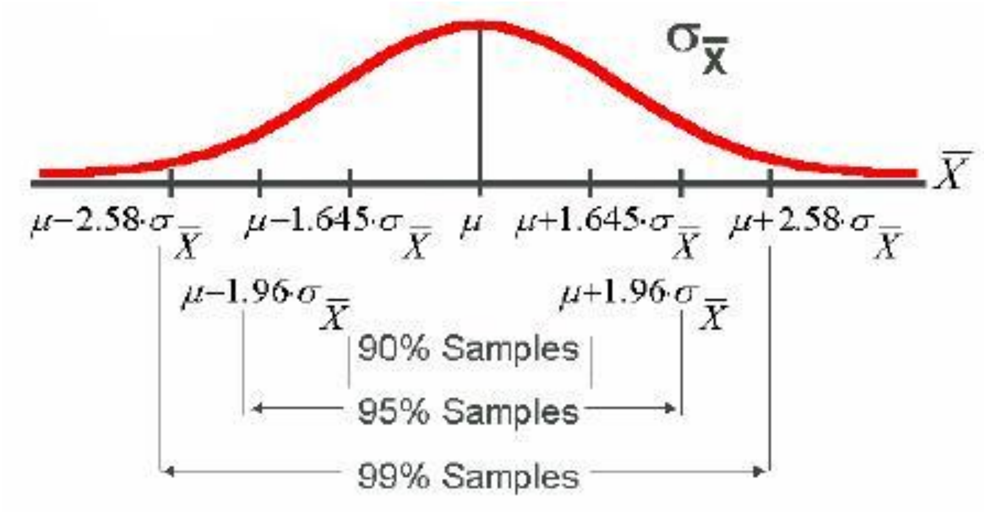
1. Z-test : $z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$

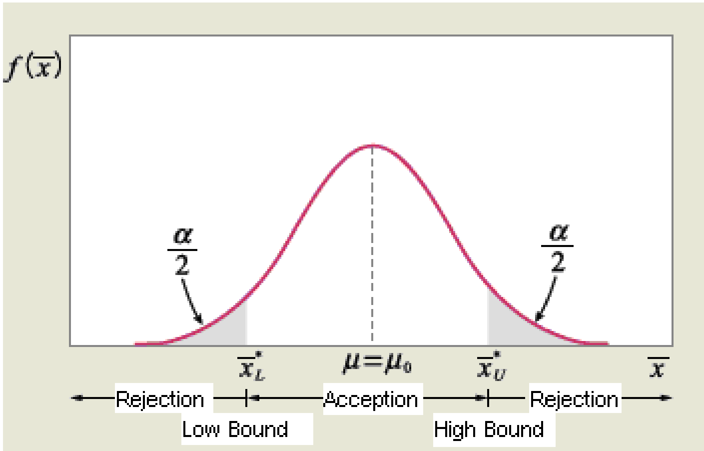
In Z-test, $\bar{x}$ is the average of the samples, μ is the polulation mean, σ is the population standard deviation, n is the sample size.

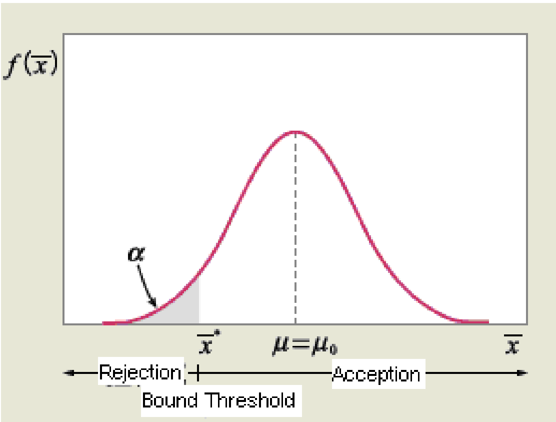
2. T-test : $t = \frac{\bar{x} - \mu}{s / \sqrt{n}}$

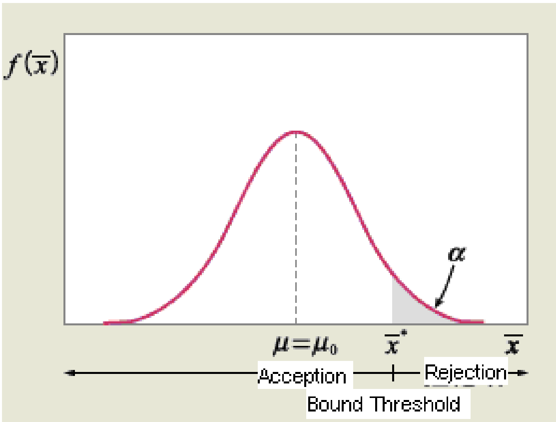
In T-test, $\bar{x}$ is the average of samples, μ is the average of controls, s is the standard deviation, n is the sample size.

During the testing, if you repeat the tests many times, the experiment results will show a normal distribution of the mean (below). To test whether the results of a particular experiment has random errors, we can use Z-test or T-test to calculate the location of the mean in the normal distribution. The researchers hypothesize the criterion to rejecte the null hypotheses (based on the percentage of area under normal distribution, or significance level), then follow the test method of Null (two-tail), RightTail (right end) and LeftTail (left tail) (below), to determine whether the results fall into the rejected experimental average range.









3. Var-Test (Chi-square variance test) : $\dfrac{(n-1)\cdot s^2}{\sigma^2}$

In Var-test, s^2 is the sample variance, σ^2 is the population variance, n is the sample size.

4. Runs-Test (Runs test of Randomness, Geary test):

$$T = T_a + T_b;$$
$$E(R) = \frac{T + T_a T_b}{T}, \quad V(R) = \frac{2T_a T_b (2T_a T_b - T)}{T^2 (T-1)}$$
$$Z = \frac{R - E(R)}{\sqrt{V(R)}}$$

In Runs-Test, T_a is the number of samples greater than the sample mean, T_b is the number of samples smaller than the sample mean. R is the number of times that a and b appear alternatively. For example, aaabbaaba, then R=5.

Properties

Hypothesis Test: There are 4 types of One-sample tests currently. We will introduce these four types of testing methods individually and the default values for all parameters are defined as follows:

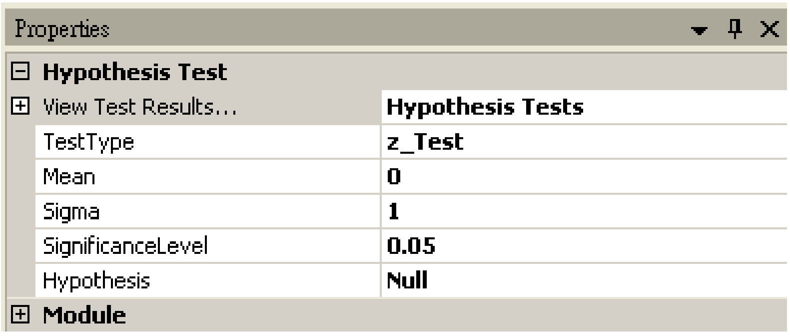
Property Name	Definition	Default
View Test Results	Use Hypothesis Test to examine test resutls	n/a
TestType	z_Test, t_Test, var_Test, runs_Test	z_Test

If you press the “View Test Result”, you will see test result as shown in the following table:

Property Name	Definition
Rejected	To show that test results fall within the range for rejection. “True” means we can reject the null hypothesis, and vice versa we can only accept the hypothesis.
SignificanceLevel	Demonstrate where the sample mean falls within the population distribution.
CI-Low	The minimum of the scope of null hypothesis, or the minimum of the confidence interval.
CI-High	The maximum of the scope of null hypothesis, or the maximum of the confidence interval.
Run Count	Refer to Introduction, this value represents the R in Runs-Test.
Above Threshold	The amount greater than the bound threshold.

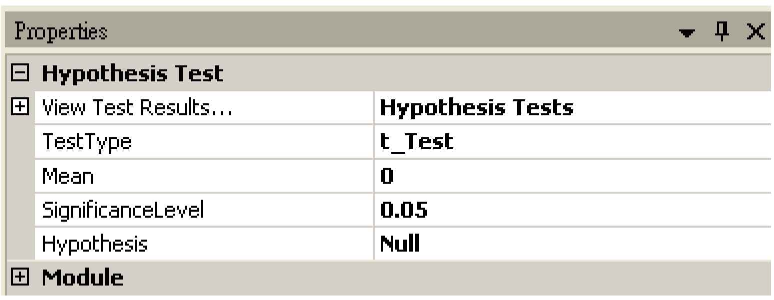
Below Threshold	The amount smaller than the bound threshold.
z-Value	This only appears in Runs-Test, the value for Z.

Z-Test



Property Name	Definition	Default
Mean	Set the average	0
Sigma	Set the standard deviation	1
SignificanceLevel	Set the criterion to reject the null hypothesis, the proportion under normal distribution. The most commonly used values are 0.1, 0.05 or 0.01. The smaller the significance level, the harder it is to reject the null hypothesis.	0.05
Hypothesis	Test methods: Null (two-tail), RightTail (right end, where the average input data must be greater than the sample mean), LeftTail (left tail, where the average input data must be less than the sample mean)	Null

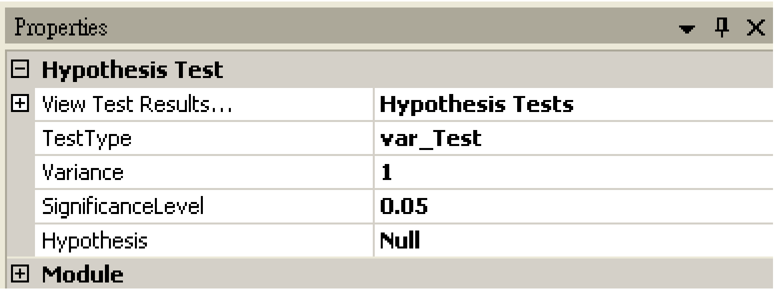
T-Test



Property Name	Definition	Default
Mean	Set the average	0
SignificanceLevel	Set the criterion to reject the null hypothesis, the	0.05

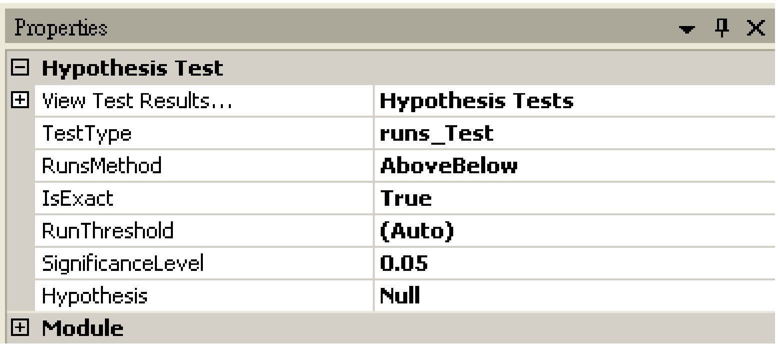
	proportion under normal distribution. The most commonly used values are 0.1, 0.05 or 0.01. The smaller the significance level, the harder it is to reject the null hypothesis.	
Hypothesis	Test methods: Null (two-tail), RightTail (right end, where the average input data must be greater than the sample mean), LeftTail (left tail, where the average input data must be less than the sample mean)	Null

Var-Test



Property Name	Definition	Default
Variance	Set the variance	0
SignificanceLevel	Set the criterion to reject the null hypothesis, the proportion under normal distribution. The most commonly used values are 0.1, 0.05 or 0.01. The smaller the significance level, the harder it is to reject the null hypothesis.	0.05
Hypothesis	Test methods: Null (two-tail), RightTail (right end, where the average input data must be greater than the sample mean), LeftTail (left tail, where the average input data must be less than the sample mean)	Null

Runs-Test



Property Name	Definition	Default
RunsMethod	AboveBelow, UpDown.	AboveBelow
IsExact	To calculate if the P-Value is calculated using the correct algorithm, this parameter only exists when RunsMethod = AboveBelow.	True
RunThreshold	Set threshold, to determine whether the information is greater than or less than the threshold. If it is “Auto”, the input data is same as the average.	Auto
SignificanceLevel	Set the criterion to reject the null hypothesis, the proportion under normal distribution. The most commonly used values are 0.1, 0.05 or 0.01. The smaller the significance level, the harder it is to reject the null hypothesis.	0.05
Hypothesis	Test methods: Null (two-tail), RightTail (right end, where the average input data must be greater than the sample mean), LeftTail (left tail, where the average input data must be less than the sample mean)	Null

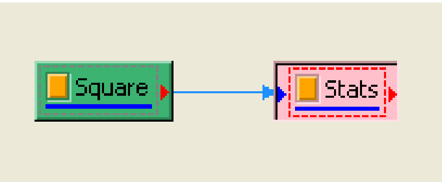
Examples

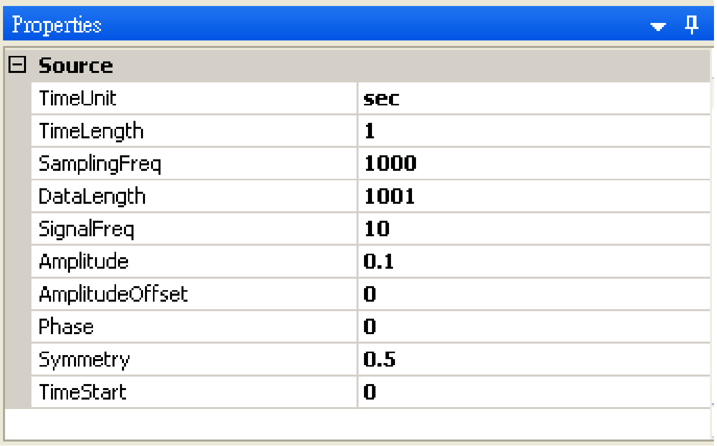
These examples will explain the use of various testing methods.

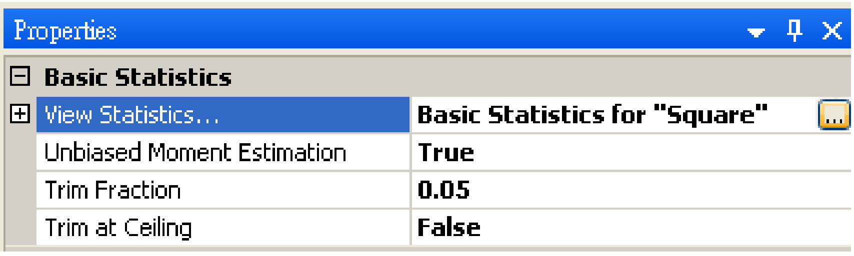
Example 1:

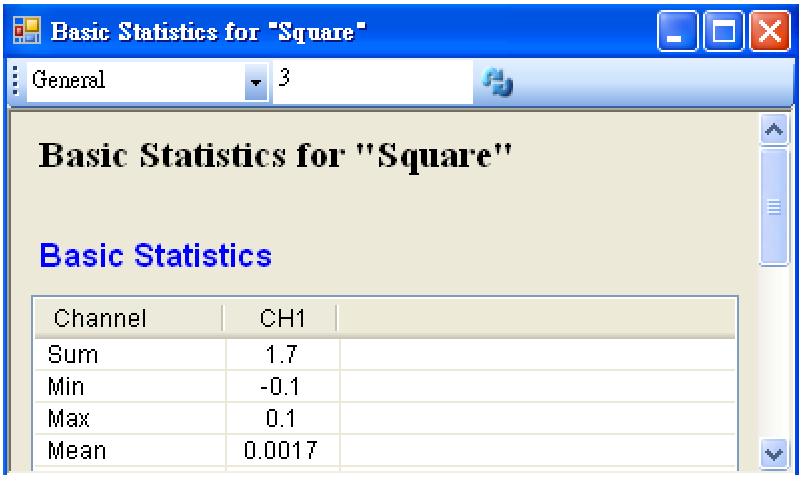
Suppose a weighing instrument, with no load, the average weight is 0.0, and the standard deviation is 0.03. Before each experiment, we will use no load to do the correction. Lets repeat reading for 1001 times, and use the information gathered to examine if the instrument has bias. First we set the null hypothesis to no error in weighing for the instrument, and then we use Z-test to test.

First use “Source/Square” to generate a group of weighing data. “Amplitude” is set to 0.1, then connect “Compute/ Statistics/Basic Statistic” Under “Basic Statistics” press “View Statistics”, observe the Mean.

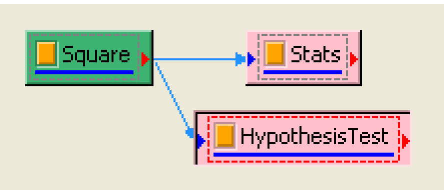


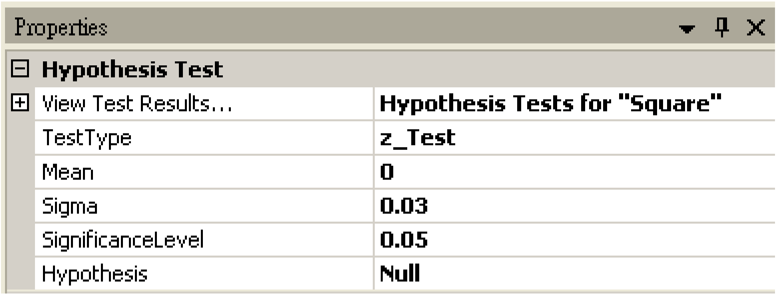


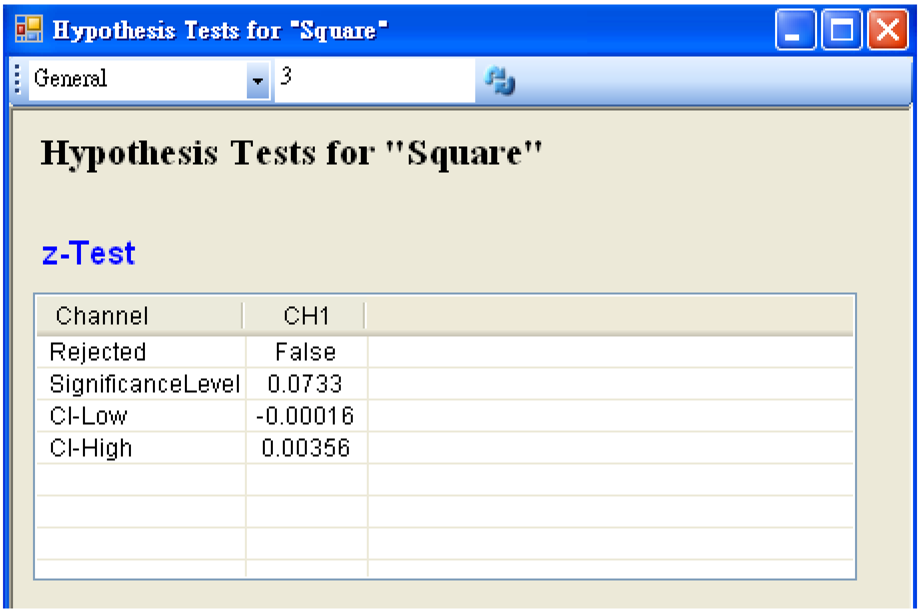




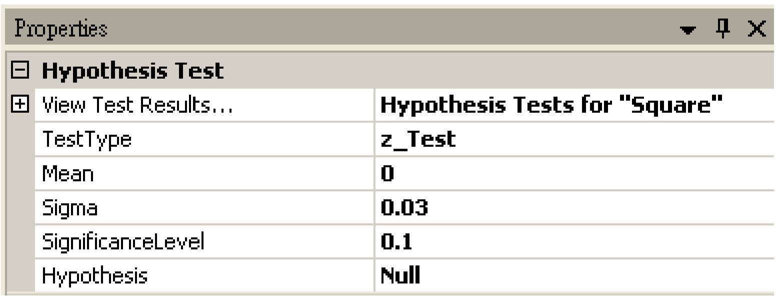
Connect “Square” to “Hypothesis Test”. Set TestType = z_Test, Sigma = 0.03, then use “View Test Results” to observe the testing results. The SignificanceLevel calculated from the data is greater than the default value; we can not reject the null hypothesis. We have to believe the null hypothesis that the instrument is not biased.

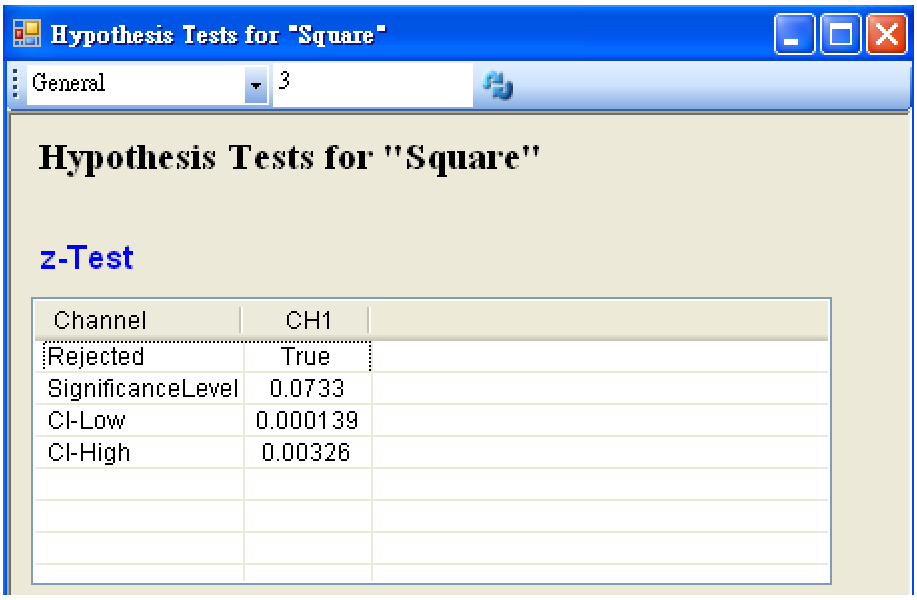






If we set the SignifianceLevle to 0.1 in the Hypothesis Test and make the range bigger to reject the null hypothesis, then use “View Test Results” to see the test result. The SignificanceLevel calculated from the data is smaller than the default value. We can reject the null hypothesis and believe that the instrument has bias and need to be adjusted.

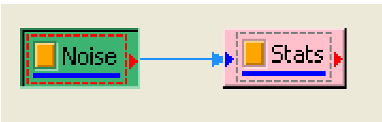


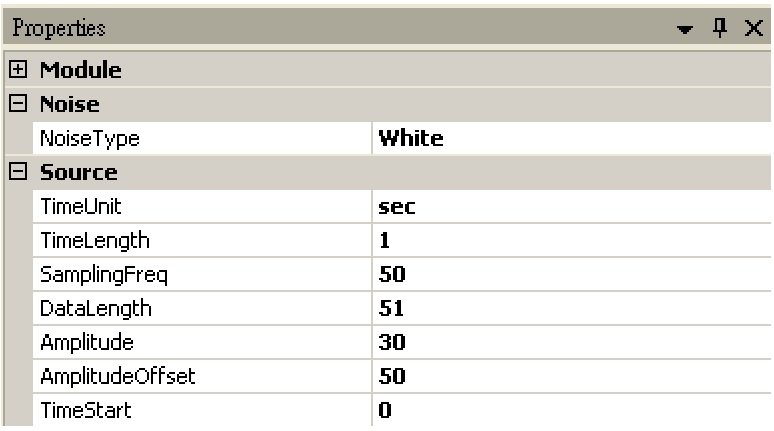


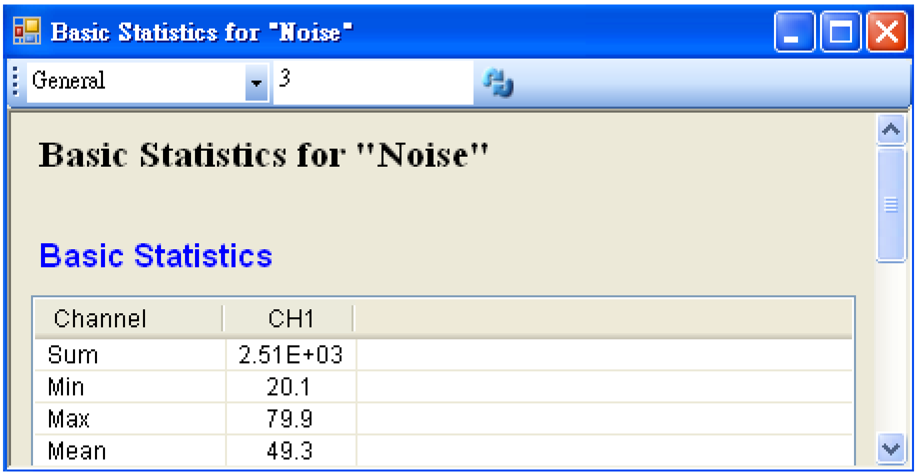
Example 2:

For example we have a class of 51 students. The head teacher is on leave, we want to know if the test score goes down because of this. The average score for the whole school is 55. We set the null hypothesis to that the test score of the class was not affected by the leave of the teacher, then preferm t-Test.

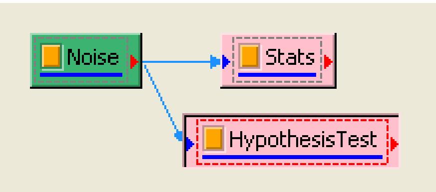
First use “Source/Noise” to generate a group of scores for the students. Set “SamplingFreq” to 50, “Amplitude” to 30, “AmplitudeOffset”to 50, then connect to “Compute/Statistics/Basic Statistics”. Under Basic Statistics press “View Statistics”, observe the Mean.

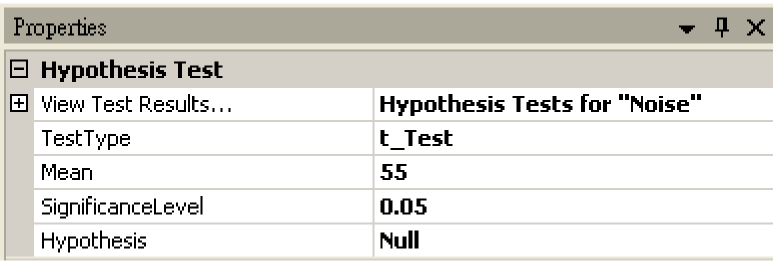


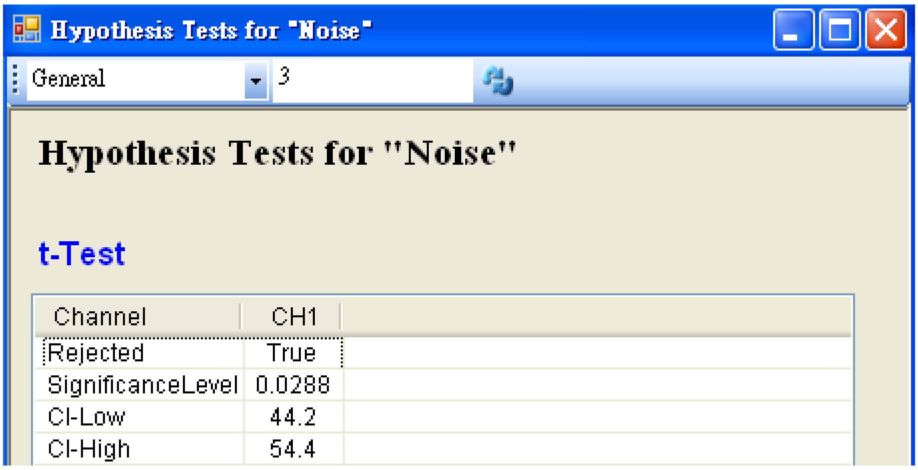




Connect Noise to Hypothesis Test, set TestType = t_Test, Mean = 55, then use “View Test Results” to check the results. The SignificanceLevel calculated from the sample is smaller than the default value, so we reject the null hypothesis and we can say that the test score for the class is affected by the teacher absence.

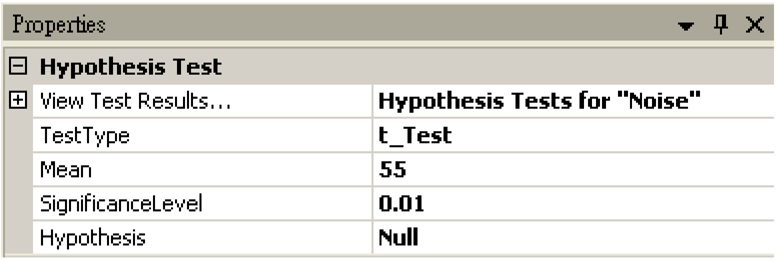


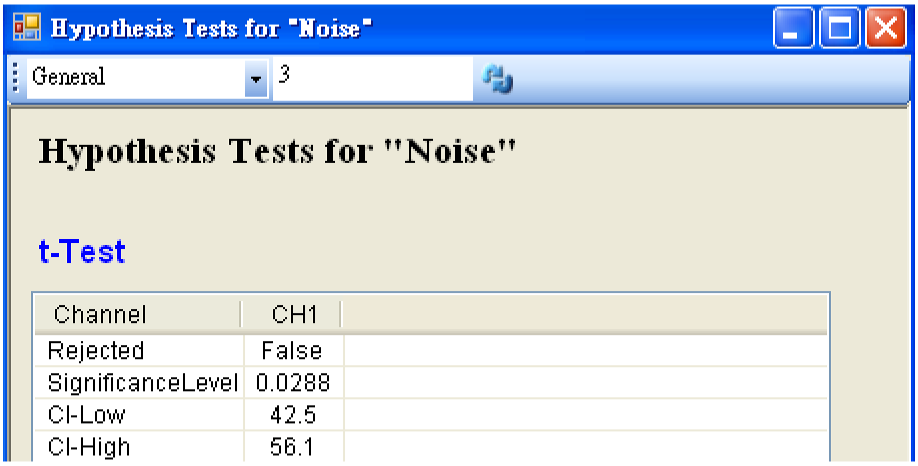




If we change the SignificanceLevle in Hypothesis Test to 0.01 to make the range of rejection of the null hypothesis smaller, then use “View Test Results” to check results. The SignificanceLevel calculated from the sample is greater than the default value,

we can not reject the null hypothesis. So we have to accept the null hypothesis and believe that the test score was not affected by teach absence.

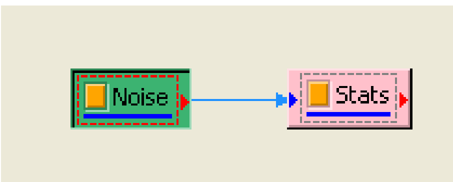


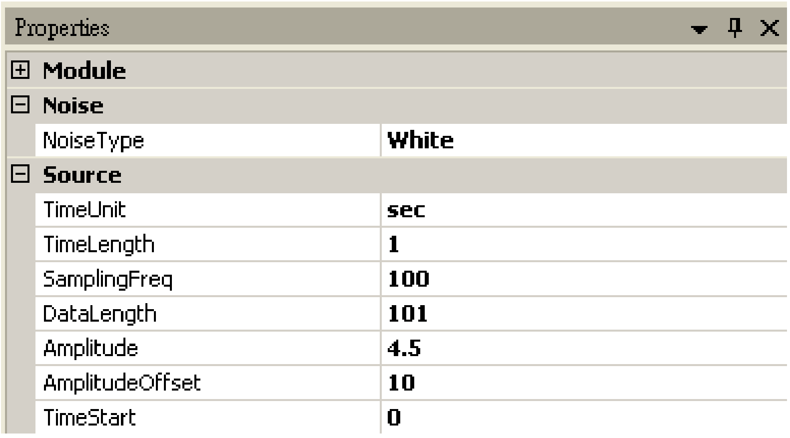


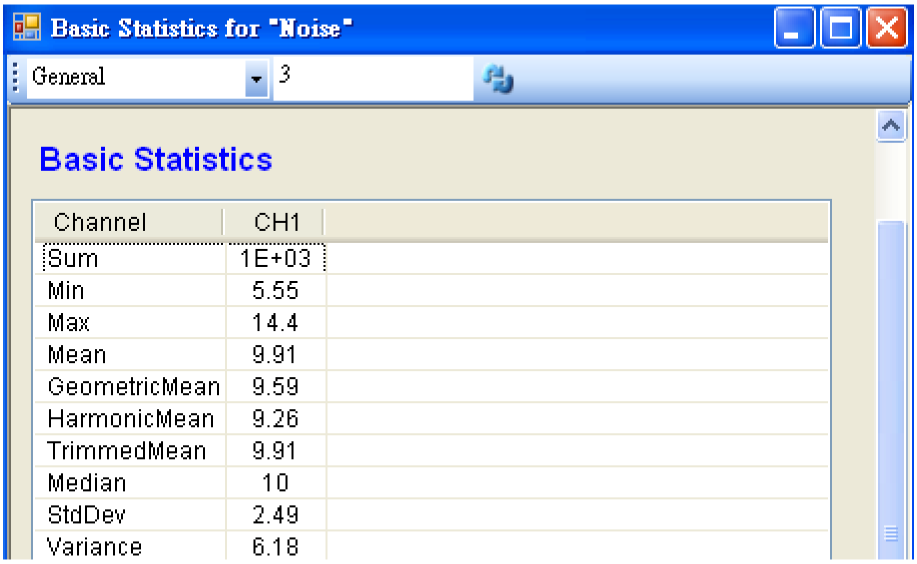
Example 3:

For example a bank has 5 windows for processing. The variance for customer waiting time variance is 10 minutes. Now we change to only have one window for processing. We will examine 101 clients to see if the variance for customer waiting time will be smaller with 1 processing window. The null hypothesis is that the variance for customer waiting time will not decrease when the number of processing window decreases. We will perform Var-Test.

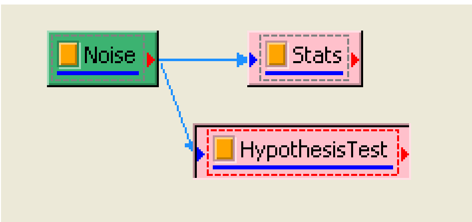
First use Source / Noise to generate customer waiting time data for 101 customers. Set SamplingFreq to 100, Amplitude to 4.5, AmplitudeOffset to 10. Then connect to “Compute/Statistics/Basic Statistic”. Under BasicStatistics press “View Statistics” to check Variance.

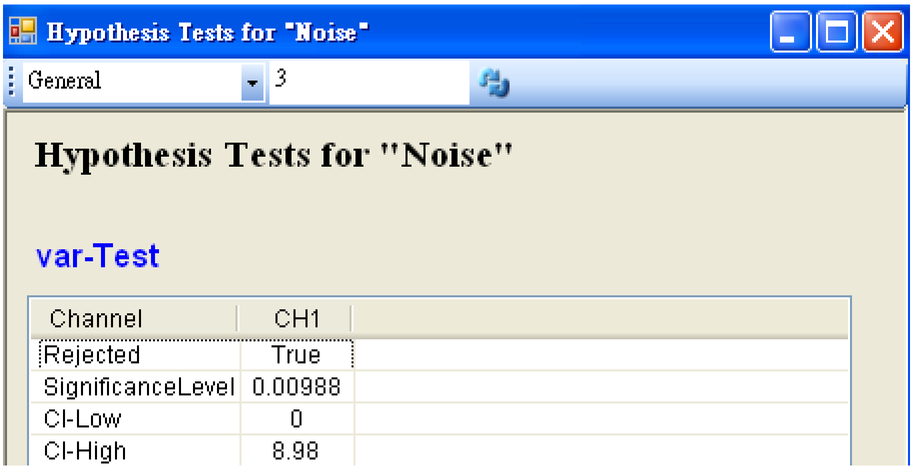






Connect Noise to Hypothesis Test, set TestType = var_Test, Variance = 10. Because the variance for samples are smaller than default, set Hypothsis = LeftTail, the use View Test Results to check results. The SignificanceLevel calculated from the samples is smaller than the default value. The null hypothesis is rejected; we can say that the variance for customer wait time is smaller under single processing window.





Related Functions

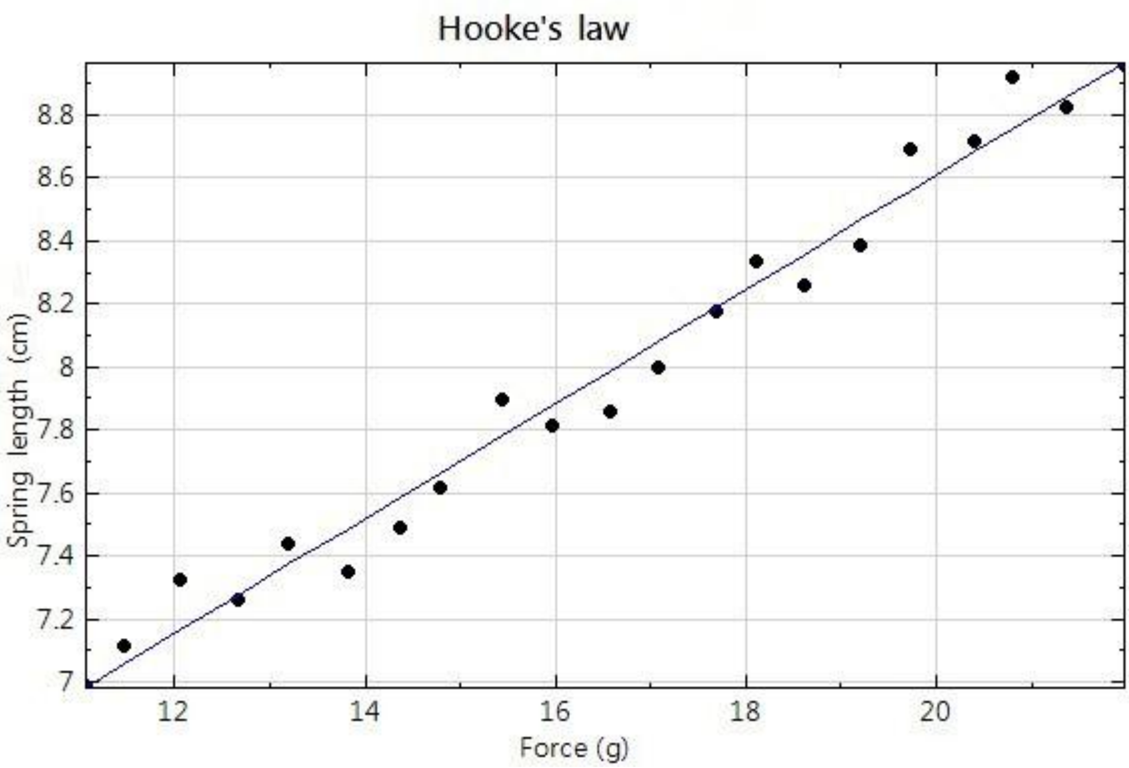
Noise, Square, Basic Statistics.

3.4.10 Linear Regression

This function is used to compute two groups of signal's linear regression line.,similar to trend line function in Excel.

Introduction

we can cite linear regression equation y =ax+b to illustrate the physical model between X and Y. The experiment to verify the Hooke's law, for instance, always measures two kinds of data, which are the weight of balance weight and the length of spring. Then use these data to compute the elastic coefficient.



The black dots represent the data derived from the measurement. The blue line is derived from the linear regression computation. The linear regression's formula is

Spring length = 0.0907* Force + 4.99

This is the relationship between the spring length and force. Compared with Hooke's law, the formula can be shown as

Force = 1/0.0907 * (Spring length - 4.99)

Therefore we can figure out that the spring's length is 4.99 (cm) and its elastic coefficient is 11.02 (g/cm). From this example we can know that it is easy to get the relationship between two groups of data by using linear regression formula. Although the data derived from measurement is discrete, some data can be indicated by linear

regression formula even if there is no measurement data. For example, from above we can know the spring length is 8.6cm when the weight is 20 g. For those unknown xj within the data range, we can get the yj by x*a+b. For those data out of the data range, we can also get them by using linear regression formula. However, the result is not that accurate since the linear formula is the simplest approximation module.

Example

This function has two types to compute linear regression：

TimeSeries (linear regression equation is x =at+b)，and

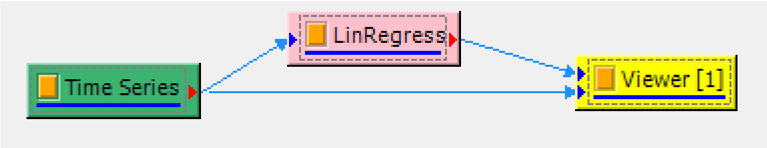
XY (linear regression equation is y=ax+b)

You can check which type you are using from Property Window. Instead of being chosen by users, the type will be changed automatically according to the formats of the input. That means users have to connect this device correctly.

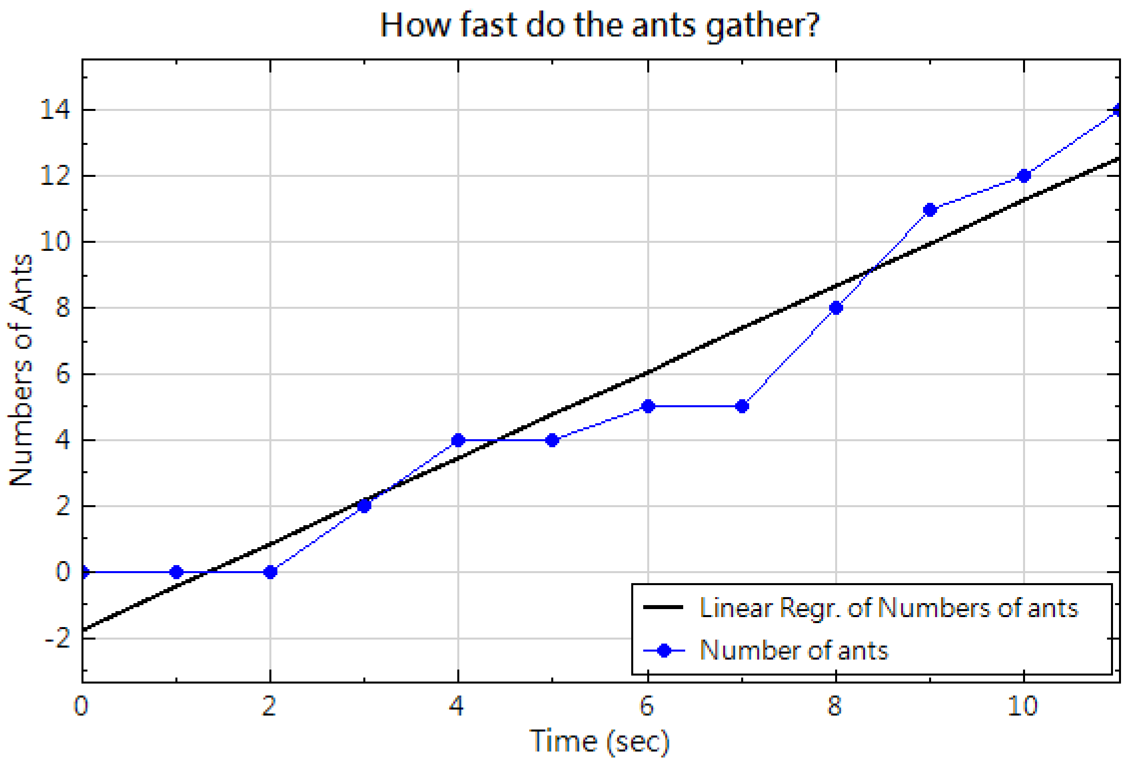
First of all, make sure that what you want to know is 1. The relationship between a group of data and time. 2. The relationship between two groups of data. Then connect the device according to the following method.

1. The relationship between a group of data and time. The corresponding format is TimeSeries.

   Connect signal and this device directly and view the result by Channel Viewer as the flow chart below.



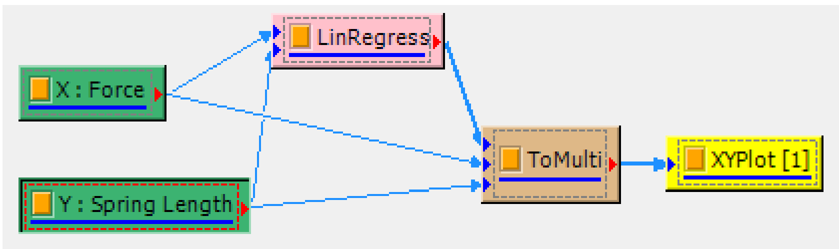
Based on the flow chart above, Viewer will draw two lines. One of them is the original time series. The other one is line after the regression. After Viewer's processing, result in the following charts.



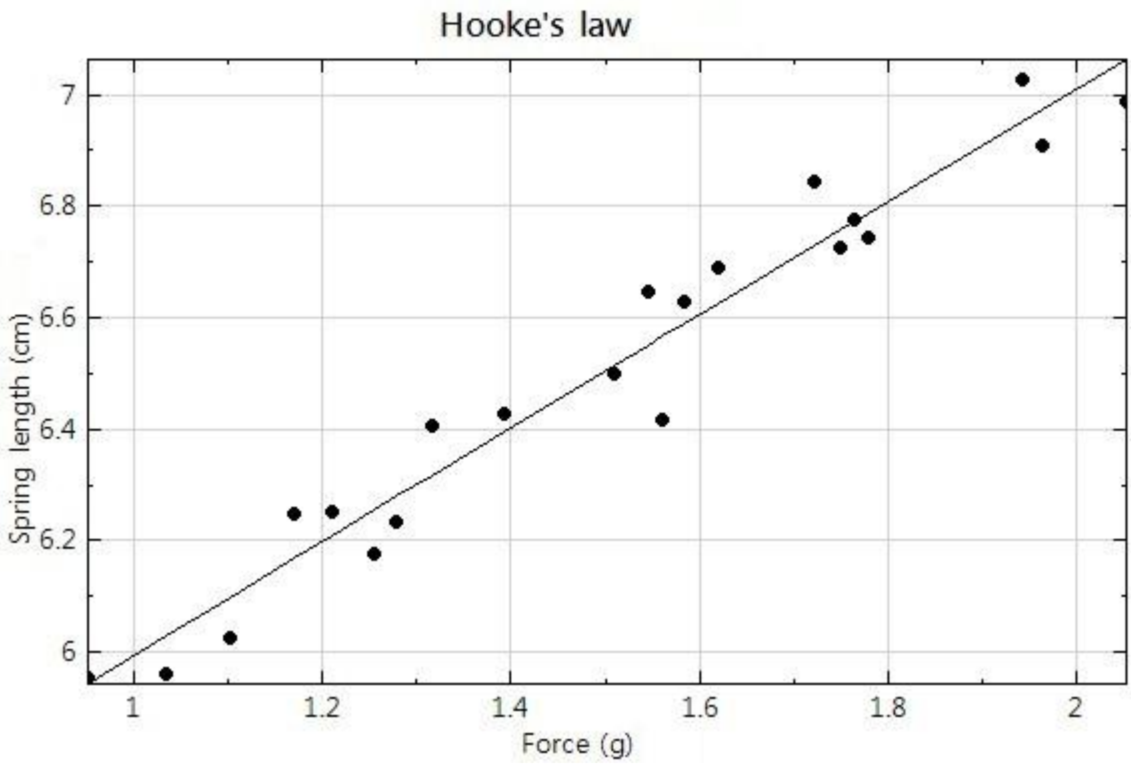
It is important to know that when it is time series, view the result by Channel Viewer, rather than XY Plot.

2. The relationship between two groups of data. The corresponding format is XY.

Connect the first group of data and the second of data with this device in order. Connect LeastSquareFit's result and data X and data Y to Merge to Multi-Channel. Connect XY Plot to view the result. This process is shown as following flow chart.



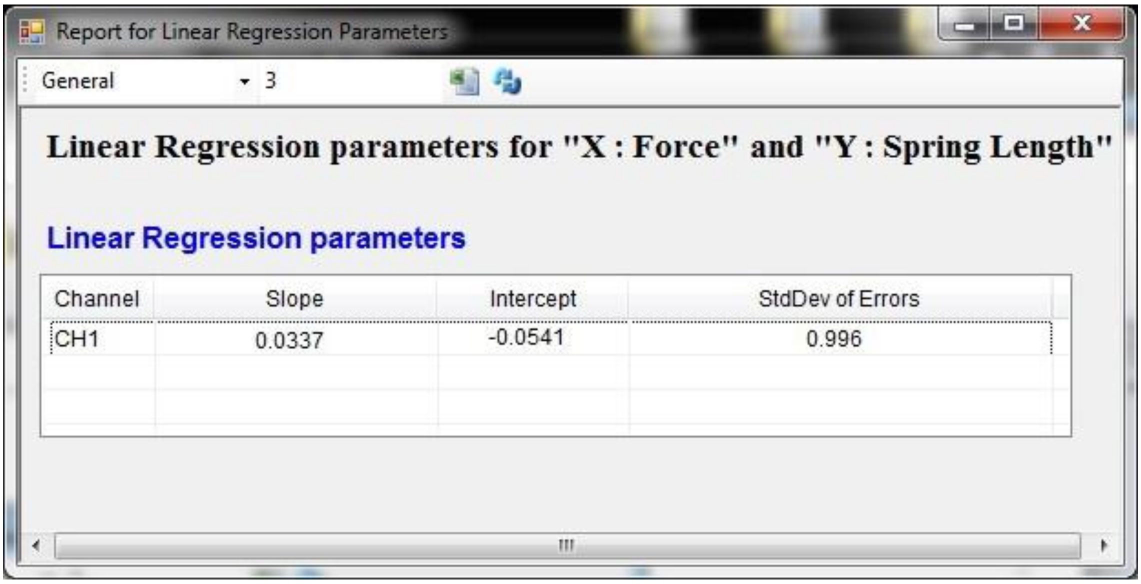
When connect to the ToMulti, the order must be Linear Regress, X , Y or X, Y, Linear Regress. The result can be shown correctly only under this circumstance. XY Plot can edit this process and the Hooke's law's chart as the illustration.



These are two fundamental ways to use this device.

Linear regression make the decision that use TimesSeries or XY based on the numbers of input. If there is only one input, Linear regression will regress every time of channel and value of this input, and then output the x of x=at+b. In case of two inputs, the first one is X and the second one is Y. After using the XY's way to regress, the outputs of regression formula are x and y. If there are multiple channels in one input, the TimeSeries can draw the regression line and original data according to what the previous paragraph depicts. But when use XY to regress, two first channels of the two inputs will regress each other. The output channel's order is x1,y1,x2,y2...etc. If we want to draw all of the X-Y data and their regression line, the chart can show in correct order only if the input order of XY Plot channel should be x1,y1,x2,y2.

In order to get the parameter of this regression line, navigate to the View Regression Parameters in Properties, and it will pop on the window below.



Then users can look up the linear parameters. As shown in the figure, Slope is the a in linear formula. The Intercept is b. Error StdDev is the mean absolute difference of every data and regression, which can be used to the reliability of the regression line.

Properties

This module accepts real number, single or multiple channel, regular signal and audio signal as input.

Property Name	Property Definition	Default Value
TimeSeries	The relationship between a group of data and time.	automatic selection, unable to change
XY	The relationship between two groups of data.	automatic selection, unable to change

Related Functions

XYPlot

3.5 TFA (Time-Frequency Analysis)

This module provides calculation of time-frequency analysis.

1. Short Term Fourier Transform
2. Morlet Transform: The generating function is Morlet function.
3. Enhanced Morlet Transform: An enhanced Morlet Transform which holds the characteristics of Short Term Fourier Transform and Morlet Transform.
4. Hilbert Spectrum: Calculate the instantaneous frequency of every time point after the input signal is processed by Hilbert Transform
5. Marginal Time/Marginal Frequency/: Perform integration on the TFA result in time/frequency spaces

3.5.1 Short-Term Fourier Transform

Short-Term Fourier Transform (STFT) is a mathematical transform related to Fourier Transform, which is used to calculate the instantaneous frequency, amplitude and phase of signals.

Introduction

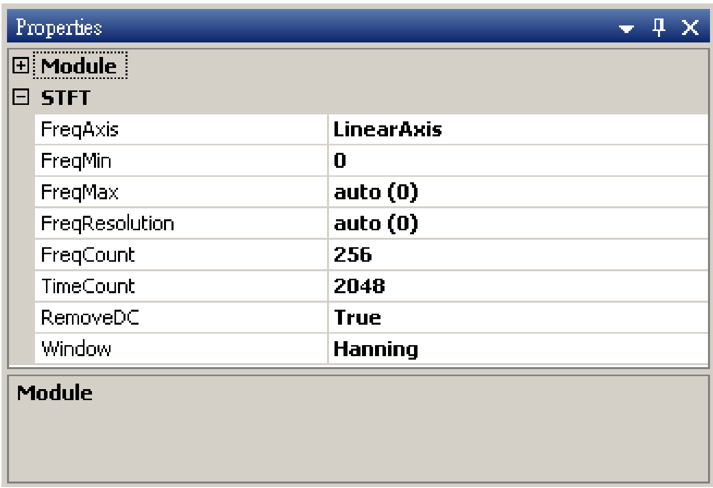
Use continuous-time function as an example, a function could multiply a time window function which is not zero, perform one-dimensional Fourier Transform, and then shift this window function along the time axis to get a series of Fourier Transform results which can be arranged to form a two-dimensional result. Mathematically, such an operation could be written as

$$STFT[x(t)] \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)\omega(t-\tau)e^{-i\omega t}dt$$

where $\omega(t-\tau)$ is the window function, $x(t)$ is the signal to be transformed. Essentially, $X(\tau,\omega)$ is a complex function obtained by performing Fourier Transform on $x(t)\omega(t-\tau)$, which represents the amplitude and phase of the input signal in time and frequency space.

Properties

This module accepts input of Signal (which could be real number, single channel, Regular) and Audio (which could be real number, single channel, Regular). The output format is complex and signal-channel spectra data.

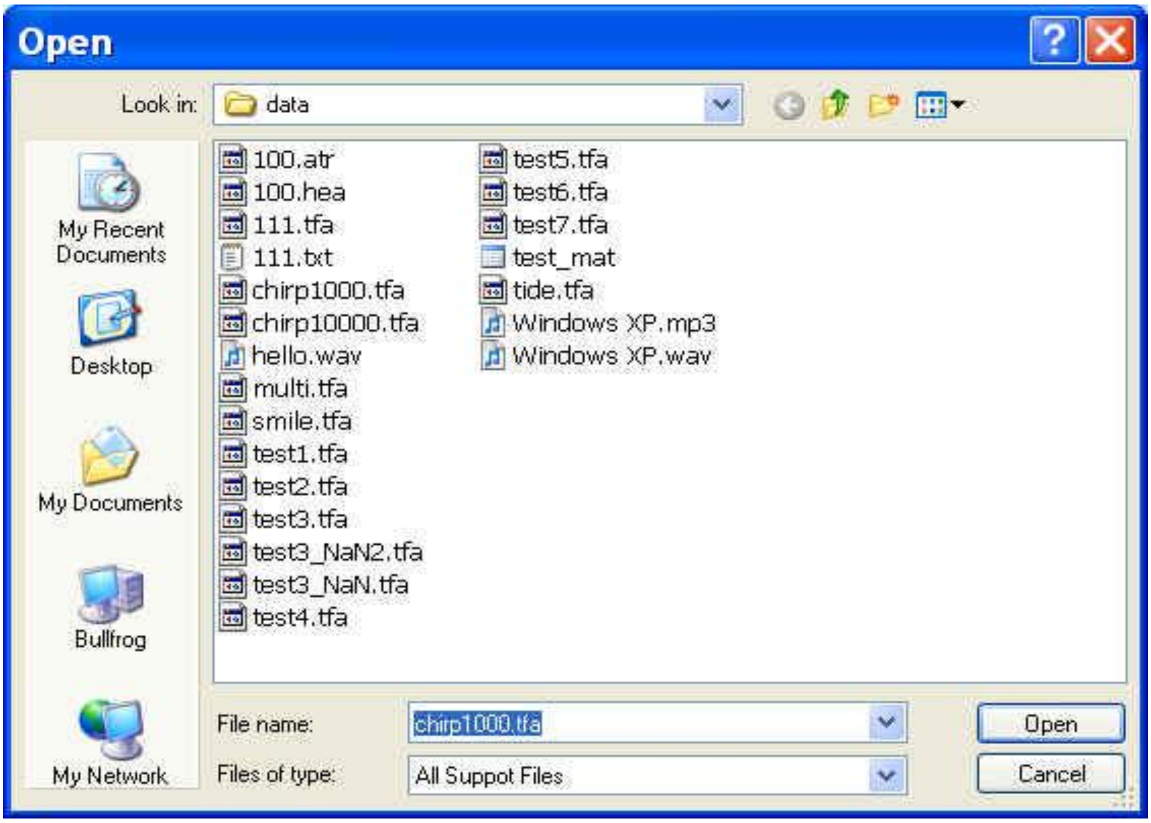


Property Name	Property Definition	Default Value
FreqAxis	The frequency axis could be LinearAxis (Linear measurement) or LogAxis (logarithmic measurement). LogAxis are mostly used in audio analysis	LinearAxis
FreqMin; FreqMax	To define the frequency boundary for frequency plotting	0; 0.5*(Sample Frequency)
FreqResolution	To define the range of the window function. It would affect the size of the window function. The smaller this value, the smaller the window function	(Sample Frequency) / 40
FreqCount	The number of discrete lattice in frequency	256
TimeCount	The number of discrete lattice in time	2048
RemoveDC	Use to choose whether remove the DC or not before STFT	True
Window	To select different window function in STFT. For the definitions of window functions, please reference to Fourier Transform	Gaussian

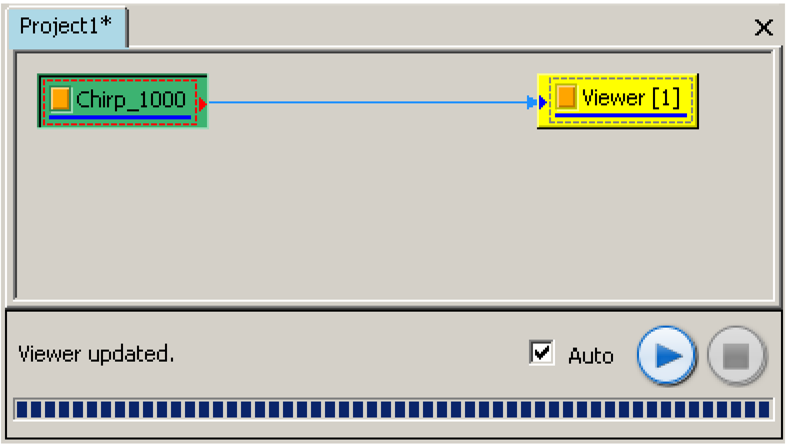
Example

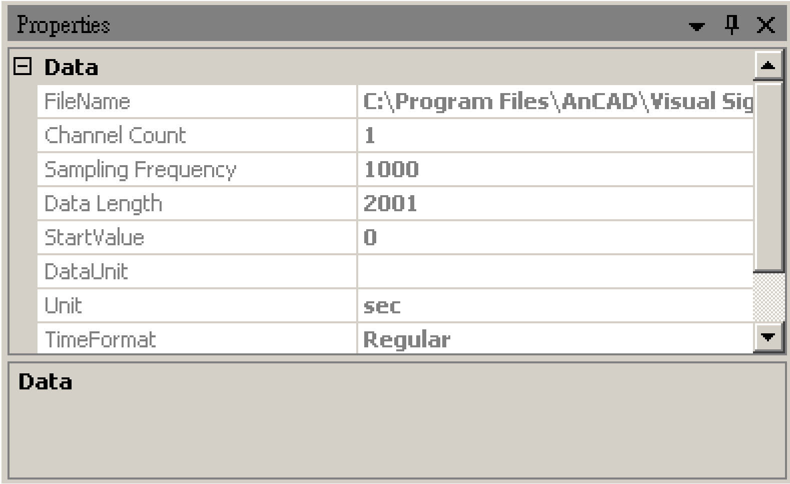
In the example below, use a Chirp signal as input, and then use DataDemon to perform time-frequency analysis. It can be seen that a frequency which varies linearly with time.

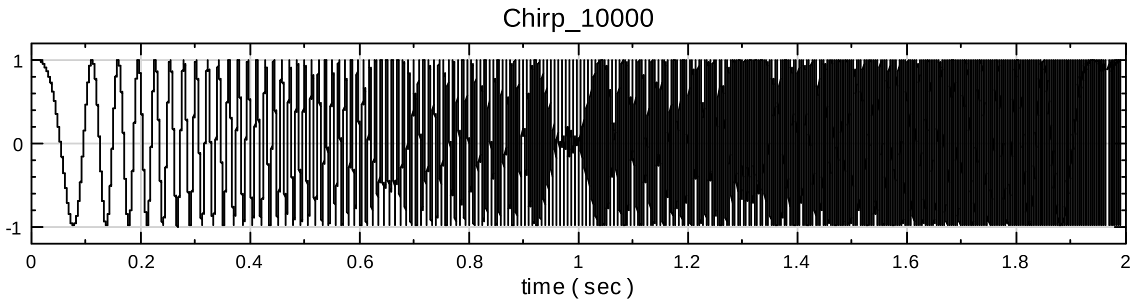
1. Press the  in the Network tools, or use *Source→Import data from file* to read a signal file, chirp1000.tfa, in the installation directory (the default directory is C:\Program Files\DynaDx\DataDemon\data)



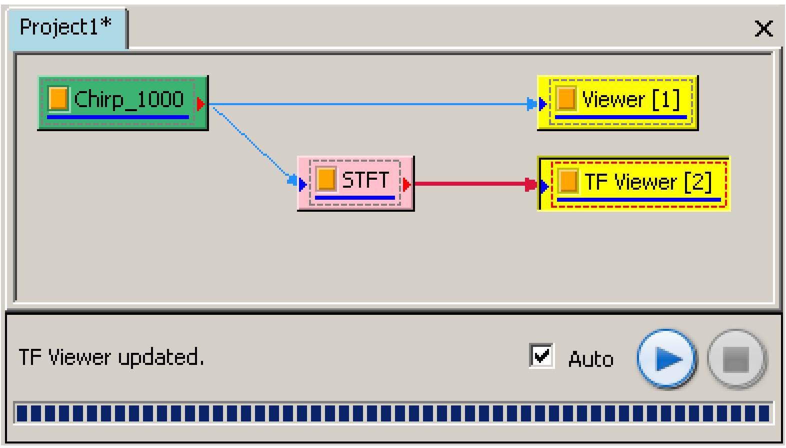
2. Click on the Chirp_1000 SFO, whose Properties show that the number of channels (Channel Count) is 1 and the sampling frequency is 1000Hz. Next, use *Viewer*→*Channel Viewer* to plot this signal. It can be seen that the signal frequency increases with the time increasing.

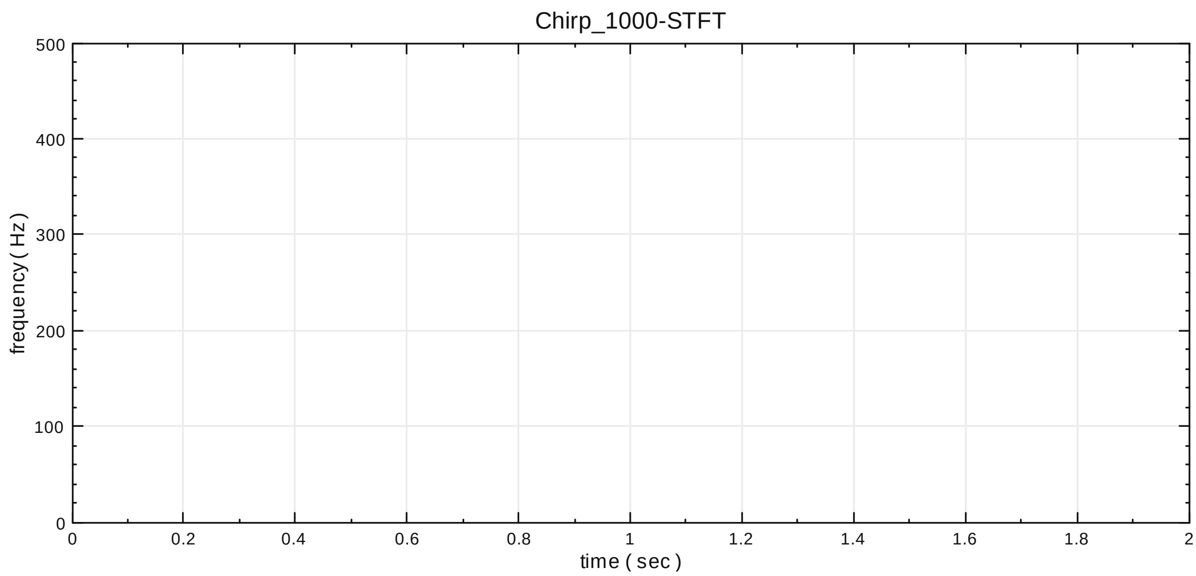




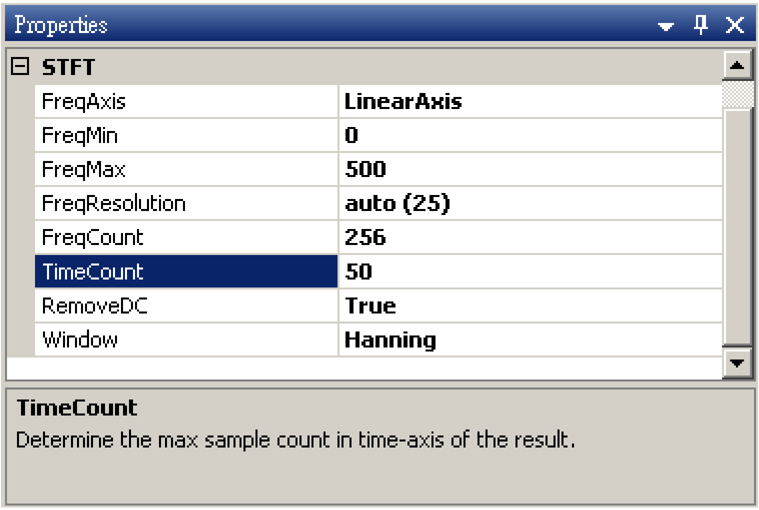


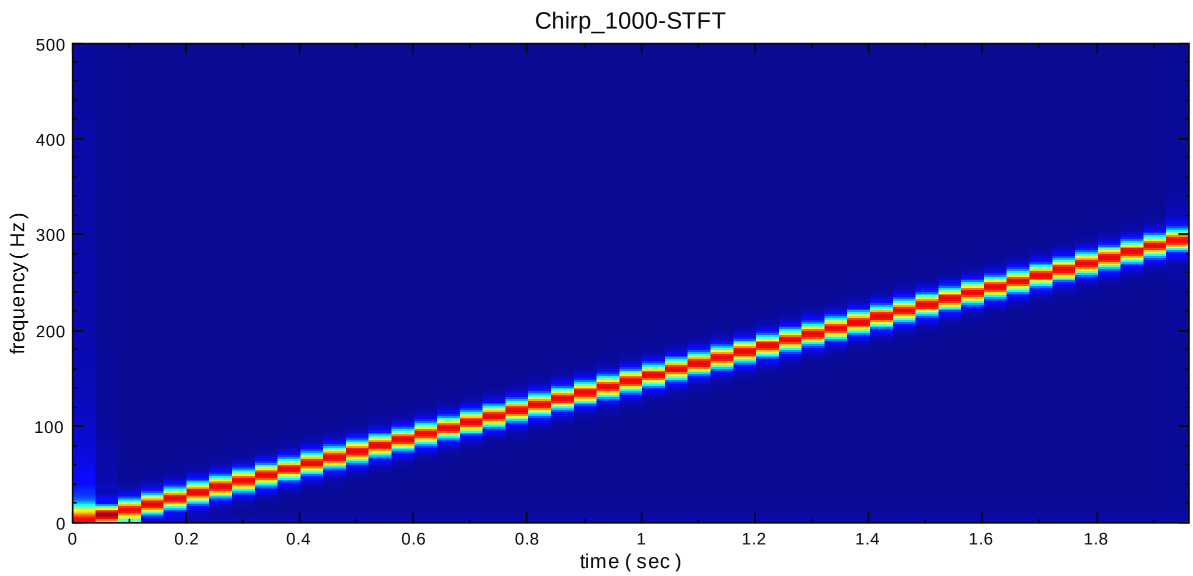
3. Select *Compute→TFA→Short Term Fourier Transform* to perform STFT on this signal and use *Viewer→Time Frequency Viewer* to plot the result. Observing the time-frequency diagram, it can be seen that the signal frequency varies lineally with time. From the result, the frequency at a given time point is available.



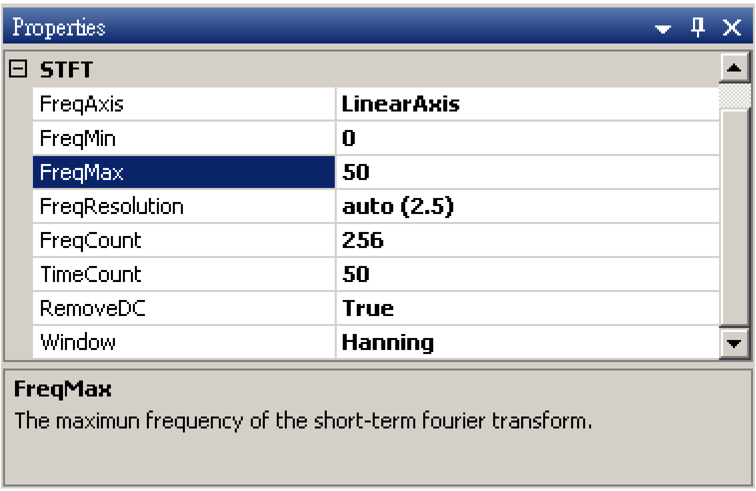


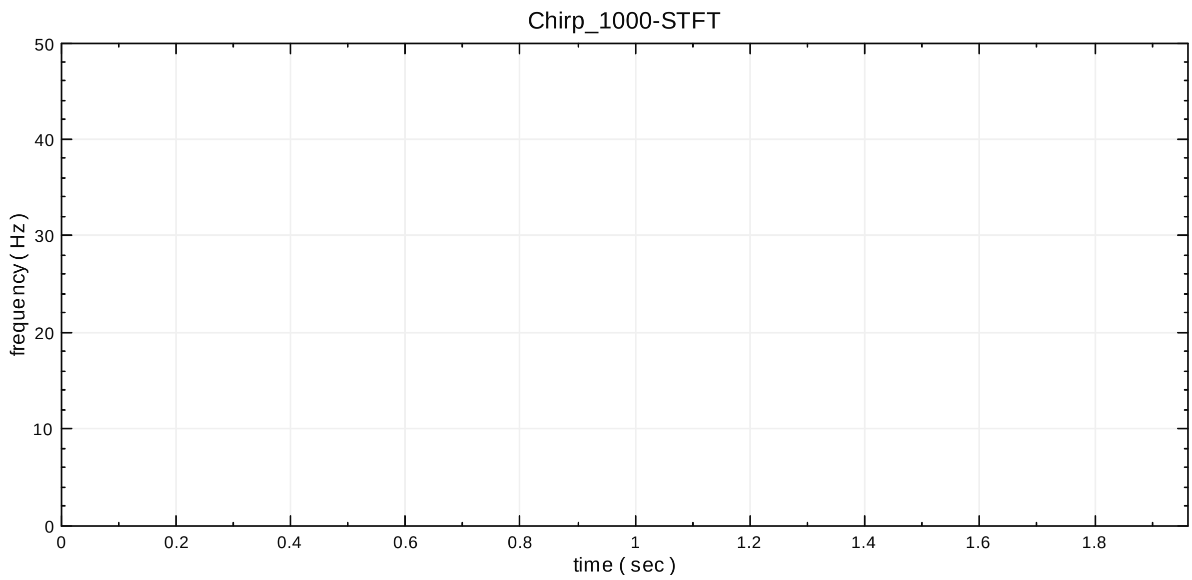
4. If the properties of TimeCount or FreqCount are changed, STFT would re-calculate based on the re-set numbers of grids. Therefore, the result resolution and computation time would be affected. Now, change the TimeCount to 50, it can be seen that the computation runs faster while the result resolution becomes worse.





5. If the properties of FreqMin or FreqMax are changed, STFT would still calculate in the original frequency range. However, it will only output the result in the range defined by the properties and would not affect the computation cost. Change the FreqMax to 50, the computation time does not decrease.





Related Functions

Fourier Transform, Morlet Transform, Enhanced Morlet Transform, Viewer.

Reference

1. A Wavelet Tour of Signal Processing (2nd Ed).

3.5.2 Morlet Transform

Wavelet Analysis, or Wavelet Transform, uses a finite-length or fast-decaying oscillating waveform, known as Mother Wavelet, to represent signals. Mother Wavelet would shrink or expand automatically based on the signal characteristics. Morlet transform uses Mother Wavelet to perform Wavelet Analysis.

Introduction

Different from Fourier Transform, the Wavelet Transform converts a signal to a time-frequency signal. Subject to the uncertainty principle, the multiplication of frequency resolution and time resolution is a fixed value, i.e., when the frequency resolution is good, the time resolution must be bad, and vise versa. The time resolution and frequency resolution of high-band and low-band frequency are fixed values in Short Term Fourier Transform. However, it is desired to have good time resolution in high-band frequency and good frequency resolution in low-band frequency. Wavelet Transform can achieve this requirement. The formula of Wavelet Transform is given below.

$$X_w(a,b)=\frac{1}{\sqrt{|(a)|}}\int_{-\infty}^{\infty}x(t)\psi\left(\frac{t-b}{a}\right)dt$$

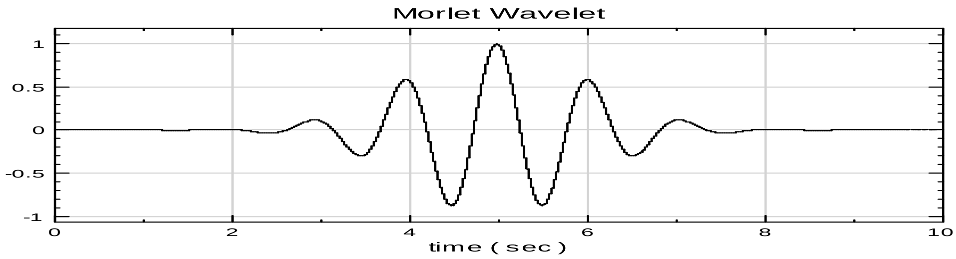
where, *a* is the scale parameter and *b* is the shift parameter of mother wavelet. Via the transform, the *a* value is converted to frequency. Mother wavelet $\psi(t)$ must satisfy 3 conditions listed below.

1. $\int_{-\infty}^{\infty}|\psi(t)|^2 dt = 1$

2. $\int_{-\infty}^{\infty}|\psi(t)|\ dt < \infty$

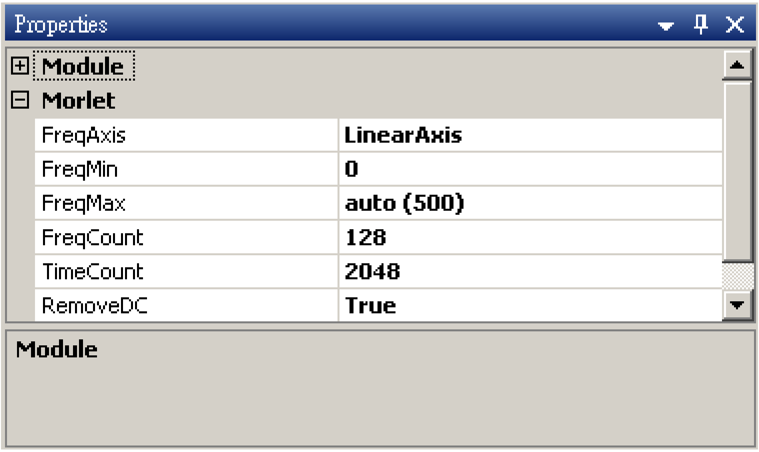
3. $\int_{-\infty}^{\infty}\psi(t)dt = 0$

Morlet Transform is one type of Wavelet Transform, whose mother wavelet definition is $\psi_\sigma(t)=e^{i\alpha t}e^{\frac{-|t|^2}{2}}$, where α is set to 6 in DataDemon.



Properties

This module accepts input of Signal (which could be real number, single channel, Regular) and Audio (which could be real number, single channel, Regular). The output format is real, signal-channel, Regular spectra data. The starting and end points must be set, and the unit is the time unit of the input signal.

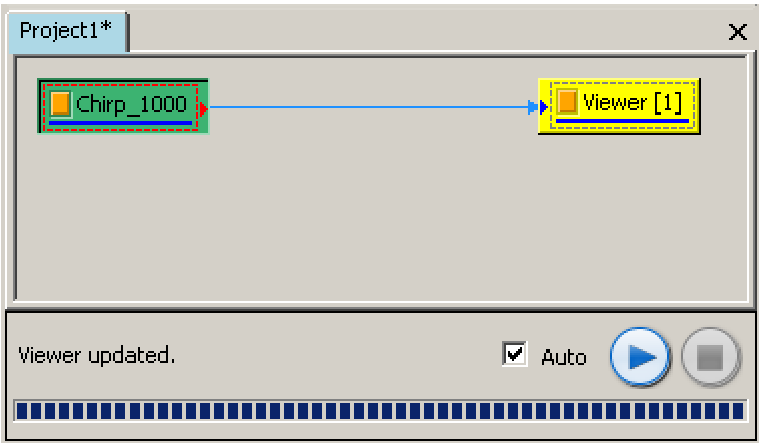


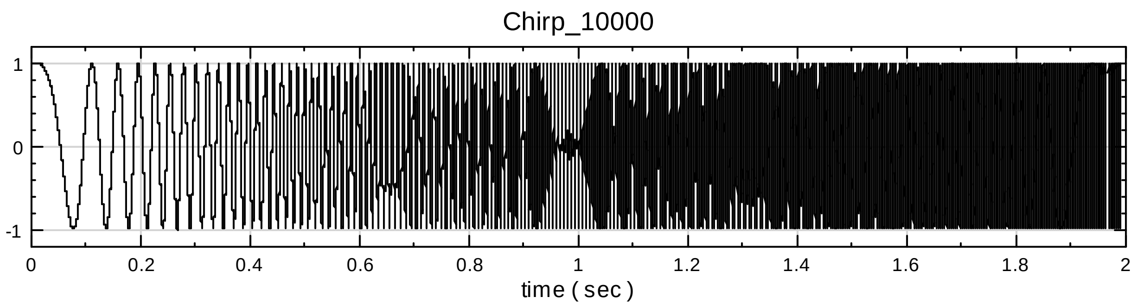
Property Name	Property Definition	Default Value
FreqAxis	The frequency axis could be LinearAxis (Linear measurement) or LogAxis (logarithmic measurement). LogAxis are used in audio analysis	LinearAxis
FreqMin; FreqMax	To define the frequency boundary for frequency plotting	0; 0.5*(Sample Frequency)
FreqCount	The number of discrete lattice in frequency	128
TimeCount	The number of discrete lattice in time	2048
RemoveDC	Use to choose whether remove the DC or not before Morlet Transform	True

Example

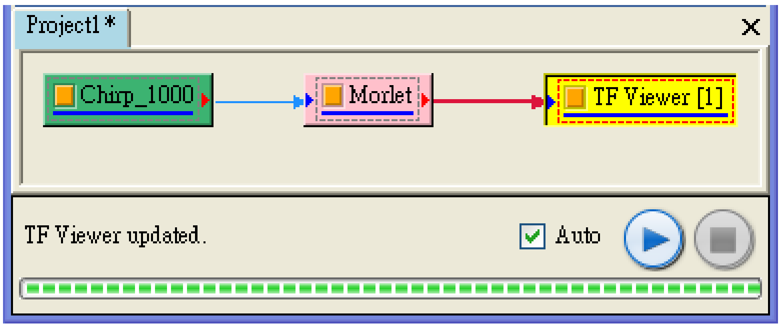
The following example is using a bird sound (Chirp) to show the change in frequency against time.

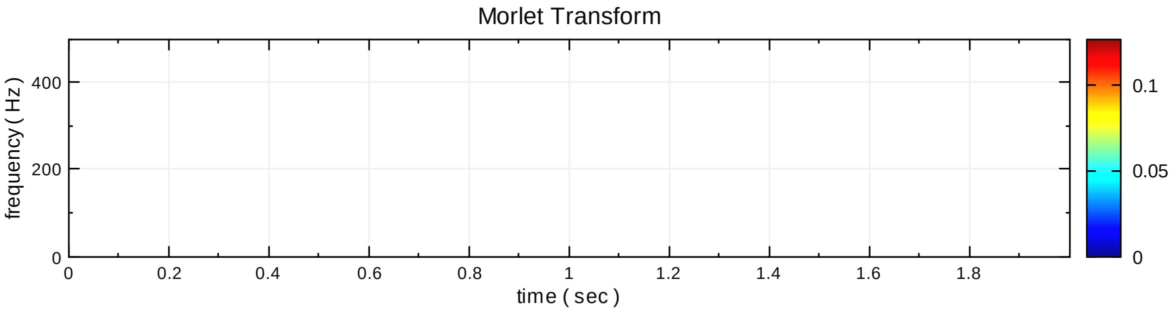
1. Press the  in the Network tools, or use *Source→Open datafrom file* to read a signal file, chirp1000.tfa, in the installation directory (the default directory is C:\Program Files\DynaDx\DataDemon\demo\basic)





2. Use *Compute→TFA→Morlet Transform* to perform calculation on the input signal and then use *TF Viewer* to plot the result. It can be seen that the higher is the frequency, the larger is the frequency spread, and therefore, the worse is the transformation performance





Related Functions

Short Term Fourier Transform, Enhanced Morlet Transform, Viewer.

References

1. A Wavelet Tour of Signal Processing (2nd Ed).
2. Y.-N. Jeng, C.-T. Chen and Y.-C. Cheng, ”The Enhanced Morlet Transform via Iterative Filter to Study Turbulent Data Strings”, The 6th Aslan Computational Fluid Dynamics Conference Taiwan, August, 2005.

3.5.3 Enhanced Morlet Transform (Professional Only)

The drawback of Morlet Transform is energy spread at high frequency due to the decrease in resolution. Please refer to Morlet Transform section for more details. Enhanced Morlet transform calculates signal with Gaussian function to resolve energy spread issue at high frequency.

Introduction

Before applying wavelet transformation, the input signal multiplies with a Gaussian function to remove the small amplitude components at the ends. So the resultion of the signal gets improved. The transform equation is shown below:

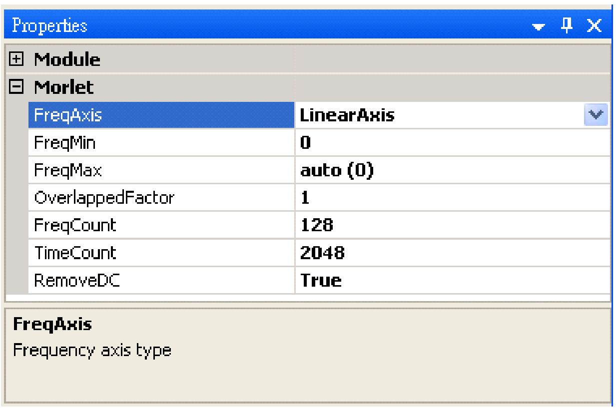
$$X_w(a,b) = \frac{1}{\sqrt{|(a)|}} \int_{-\infty}^{\infty} x(t)\psi\left[\frac{t-b}{a}\right] G(\sigma,b,t)\,dt$$

where $G(\sigma,b,t)$ is the Gussian function $\left[\frac{1}{4\pi}\cdot\frac{1}{\sigma}\right]^{\frac{1}{2}} \cdot e^{\frac{1}{4\sigma}(t-b)^2}$

At high frequency, the parameter a of the Scale in Morlet wavelet becomes smaller; this reduces the resolution at high frequency. Before the transformation, Morlet Wavelet multiplies Gaussian Window $G(\sigma,b,t)$ to improve the resolution at high frequency.

Properties

This module accepts input signals of real number, single channel, regular, and audio.The format of the output is real, single channel, and the spectra of regular. Properties are defined below.



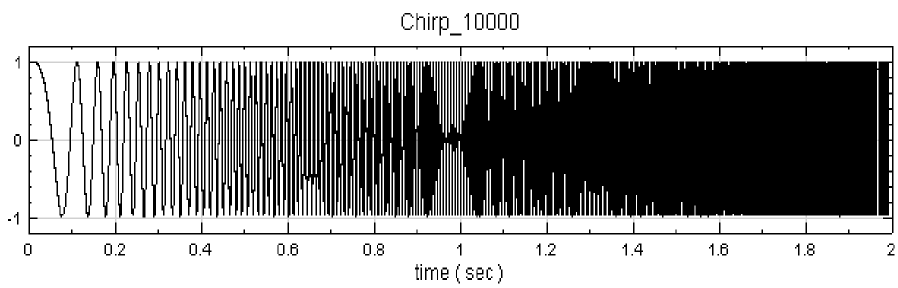
Property Name	Property Definition	Default Value
FreqAxis	The frequency distribution of spectra can	LinearAxis

	be in Linear scale or Log scale.	
FreqMin ; FreqMax	Set the frequency range, minimum and maximum	0; 0.5 * (Sample Frequency)
OverlappedFactor	The overlapping factor when moving Gaussian window, it is σ in the equation	1
FreqCount	Set grid count in frequency axis	128
TimeCount	Set grid count in time axis	2048
RemoveDC	Remove the DC component before performing Morlet Transform	True

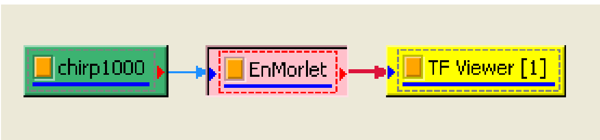
Example

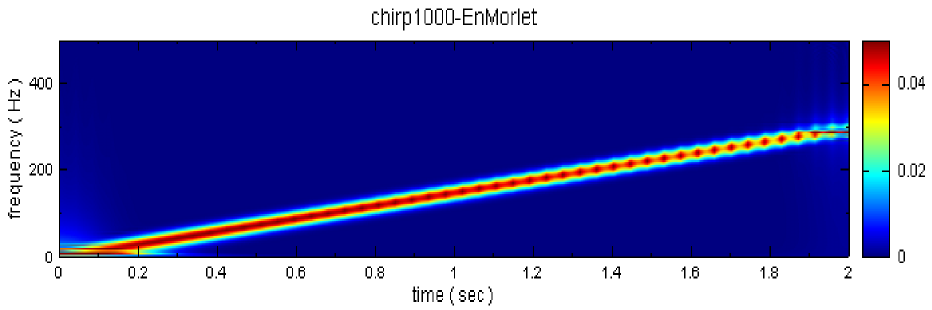
The input signal is chirp sound of birds. We calculate its spectra and show that its frequency varies with time.

Click button in Network tools, or use Source / Open data from file to read signal from a file. The file is chirp1000.tfa and it is located at demo directory of the installation. (The default location C:\Program Files\DynaDx\DataDemon\demo\Basic), then show the data with Viewer / Channel Viewer.



Do the transform with Compute / TFA / Enhanced Morlet Transform and show the result with TFA Viewer. It shows that the result looks better than Morlet Transform at high frequency.





Related Functions

Short Term Fourier Transform, Morlet Transform.

References

1. A Wavelet Tour of Signal Processing (2nd Ed).
2. Y.-N. Jeng, C.-T. Chen and Y.-C. Cheng, "The Enhanced Morlet Transform via Iterative Filter to Study Turbulent Data Strings", The 6th Aslan Computational Fluid Dynamics Conference Taiwan, August, 2005.

3.5.4 Hilbert Spectrum

Hilbert Spectrum is a time-frequency matrix outputed from Hilbert Transform. Its elements are instaneous frequency and instaneous amplitude. For HHT, EMD method extracts IMFs from the input signal. Then Hilbert Transform can be applied to each IMF for Hilbert Spectrum results. The output of Haar Transform can directly apply to Hilbert Spectrum also.

Introduction

Please refer to Hilbert Transform.

For input signal $x(t)$, we can calculate its Hilbert Transform $y(t)$ as following:

$$H[x(t)] = y(t) = \frac{1}{\pi}P.V\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}d\tau$$

where $y(t)$ is called Hilbert pair of $x(t)$. The above equation is equavilent to the convolution between $x(t)$ and $\frac{1}{t}$, devide by π, i.e. $\frac{1}{\pi}\cdot x(t)*\left(\frac{1}{t}\right)$. And *P.V* is Cauchy Principle Value.

$$z(t) = x(t) + iy(t) = a(t)e^{i\theta(t)}$$

$$a(t) = \sqrt{x^2+y^2}$$

$$\theta(t) = tan^{-1}\frac{y(t)}{x(t)}$$

Define $\omega(t) = \frac{d\theta(t)}{dt}$ as the instaneous angle speed, and $\frac{1}{2\pi}\omega(t)$ is the instaneous frequency.

So after Hilbert transform, we can obtain $y(t), z(t), \omega(t)$.

To plot the time-frequency graph, $a(t), \omega(t)$ varies based on time t,

When the time is t, the height along Y axis can be obtained from frequency $\frac{\omega(t)}{2\pi}$.

When the time is t, the amplitude along Z axis can be obtained from $|a(t)|$.

Once the calculation is done for all data points, the time-frequency graph completes. This is Hilbert Transform. Since the graph obtained from discrete data points, Gaussian function can be applied to smooth the curve.

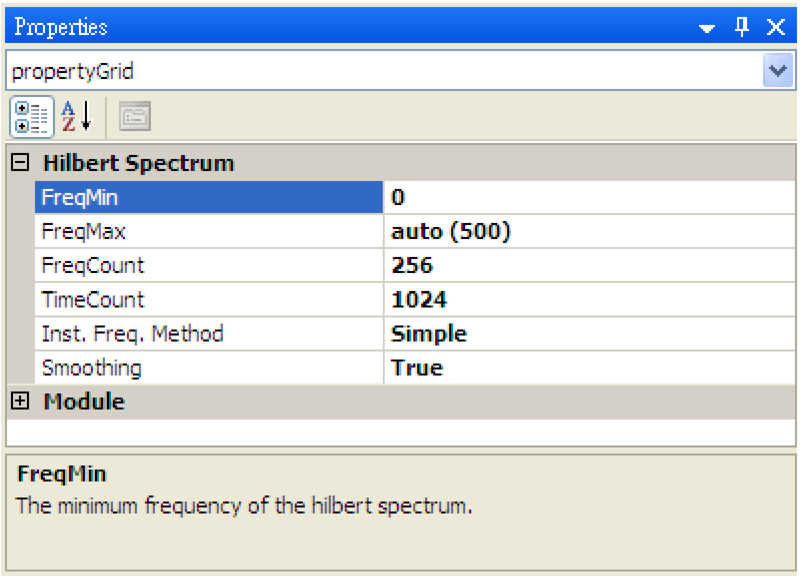
If all of the multi-channel signal are applied with Hilbert transform, the original signal can be represented as following (mostly apply to the results of EMD calculation):

$$\sum x_j(t) = Re\left(\sum a_j(t) e^{i\int \omega_j \cdot dt}\right)$$

The rest of the operations are the same as the one for single-channel signal.

Properties

This module accepts input signal of real number, single channel or multi-channel, regular, and audio. The format of the output is real, single channel, spectra of regular data. The properties are defined below.

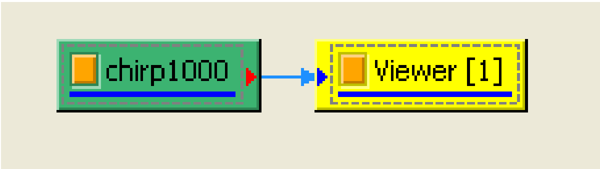


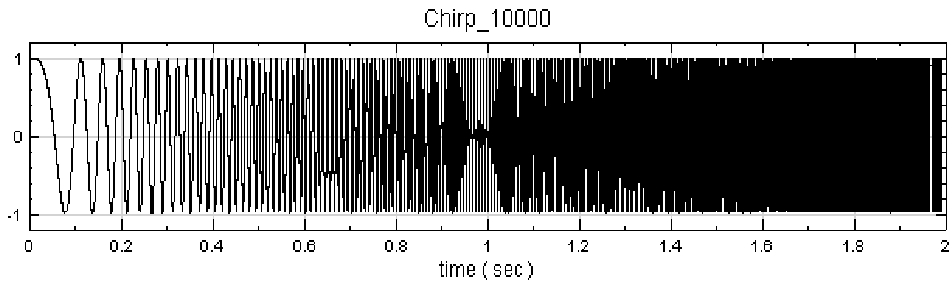
Property Name	Property Definition	Default Value
FreqMin	Set display minimum frequency of Hilbert Spectrum	0
FreqMax	Set display maximum frequency of Hilbert Spectrum	0.5 * (Sample frequenca)
FreqCount	Set grid count of Hilbert Spectrum along frequency axis	256
TimeCount	Set grid count of Hilbert Spectrum along time axis	1024

Inst. Freq. Method	Method to calculate instaneous frequency: Simple or Barnes. Please refer to Hilbert Transform.	Simple
Smoothing	Apply Gaussian function to smooth the curve	True

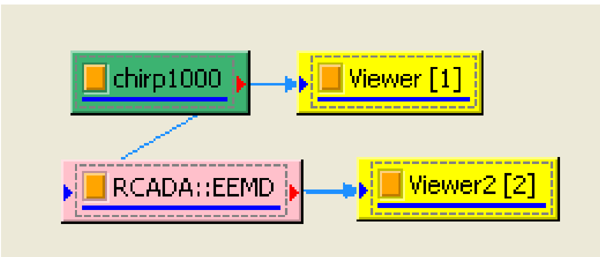
Example

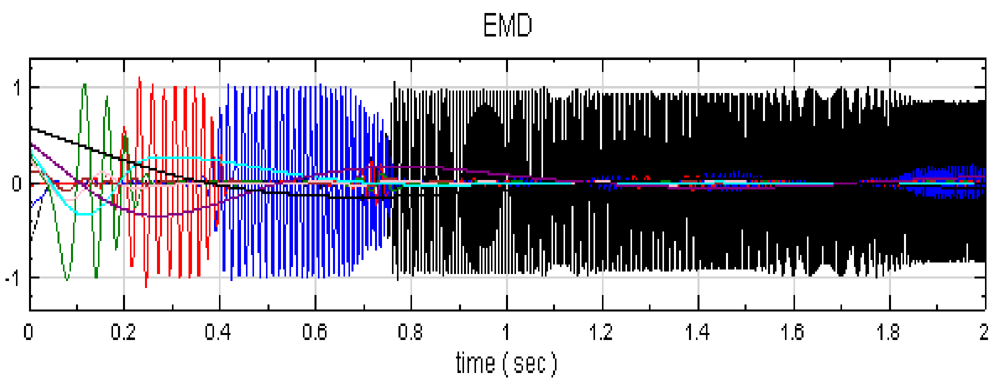
Click  button in Network tools, or use Source / Open data from file to read signal from a file. The file is chirp1000.tfa and it is located at demo directory of the installation. (The default location C:\Program Files\DynaDx\DataDemon\demo\Basic), then show the data with Viewer / Channel Viewer.



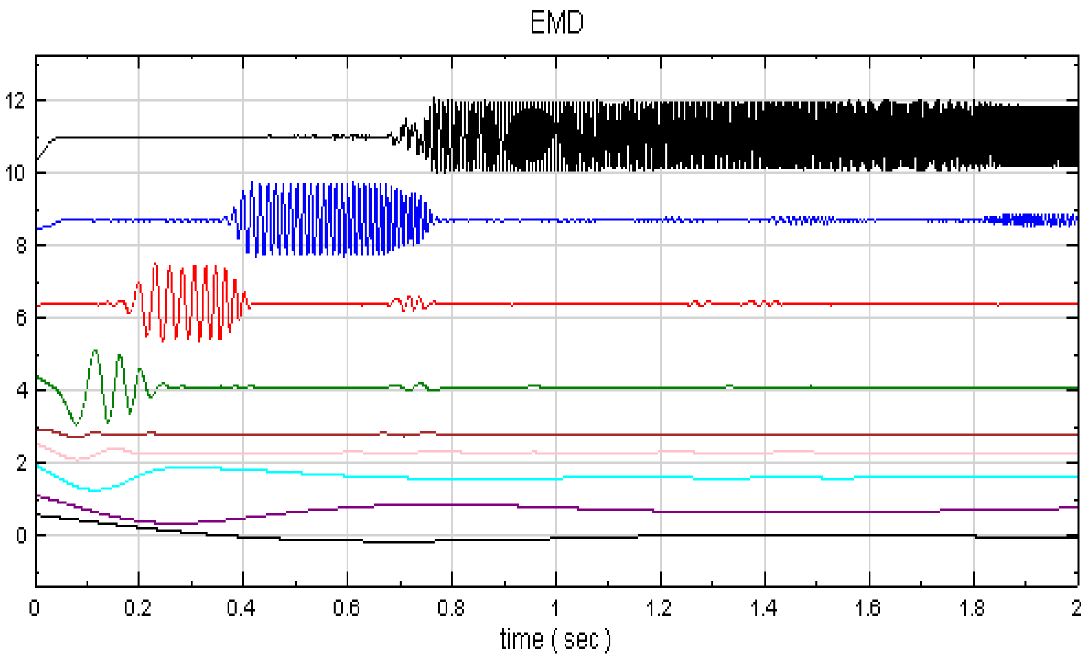


Connect Chirp_1000 to Compute / HHT / RCADA EEMD for calculating IMF (Intrinsic mode function) and display the results using Viewer / Channel viewer.

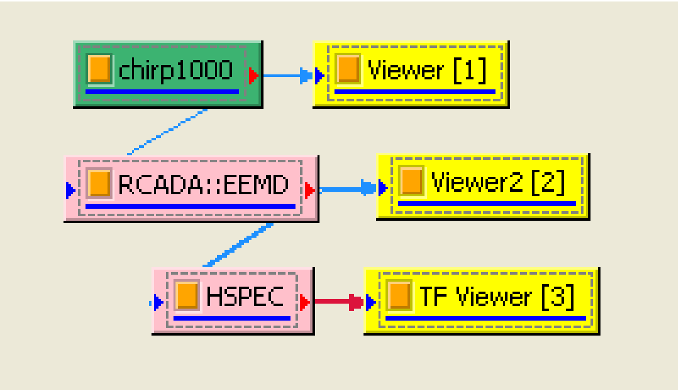


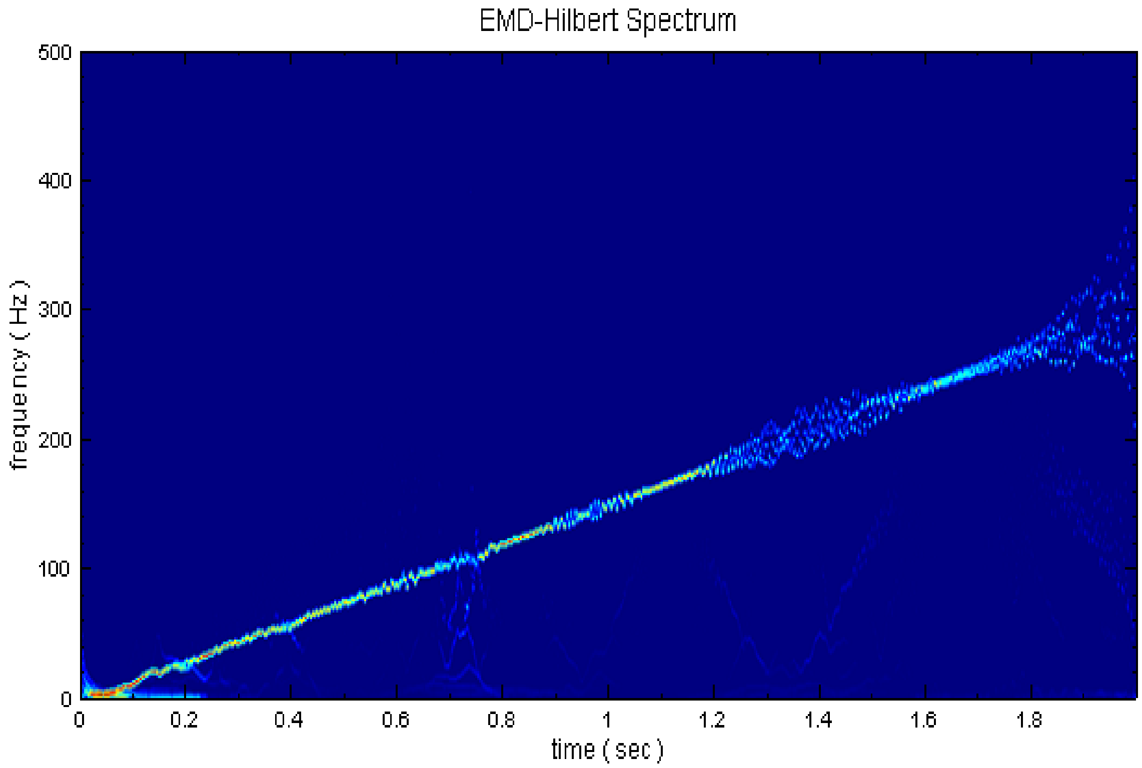


Since the result of EMD is multi-channel, the above graph is hard to see. For better viewing, change the setting of Channel viewer, set Properties / ViewerHeight to 300 and set Multi-channeldisplay to List.

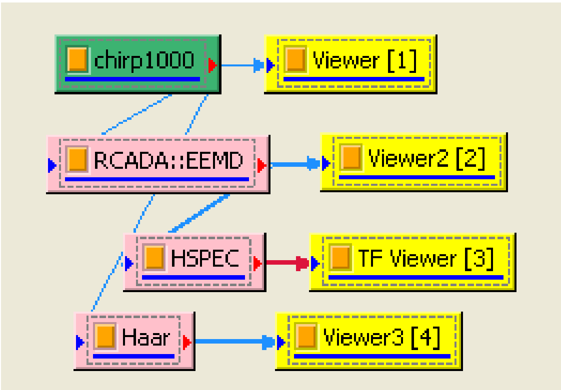


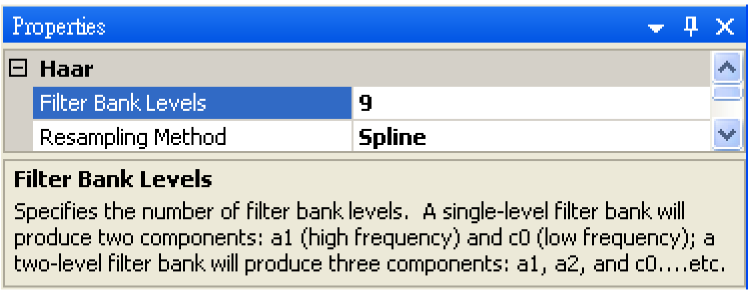
Connect RADAC EEMD to Compute / TFA / Hilbert Spectrum, and then to Viewer / Time-frequency viewer for displaying results. This is how to show the instaneous frequency of HHT.

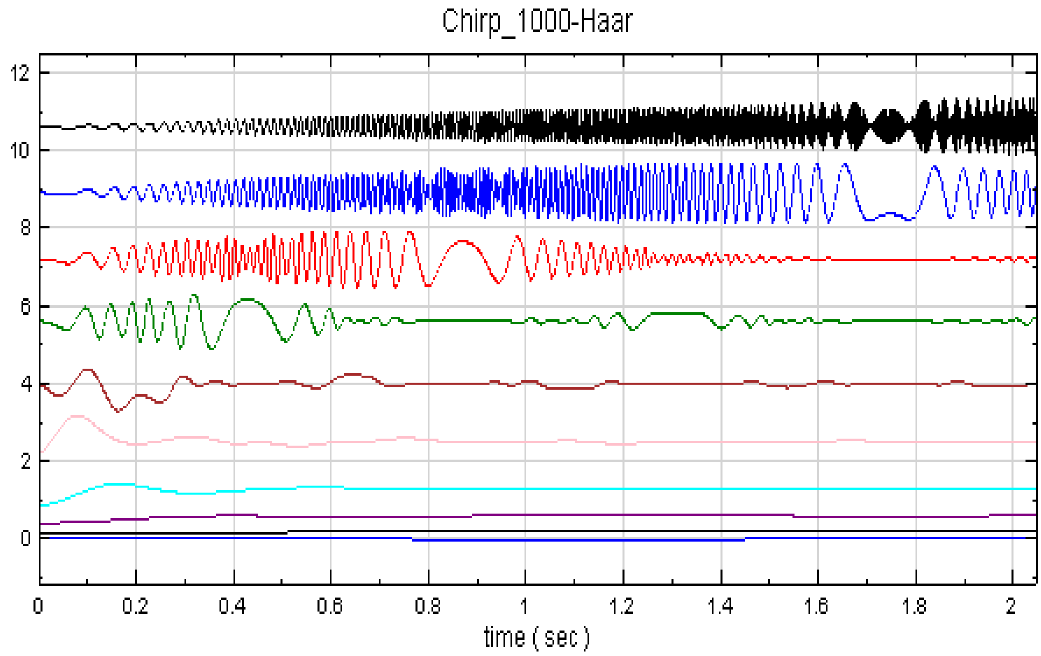


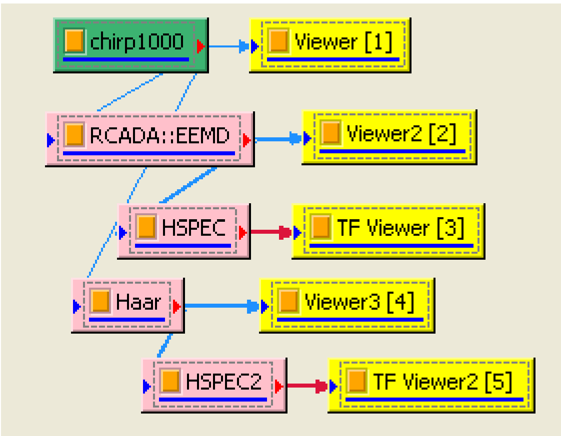


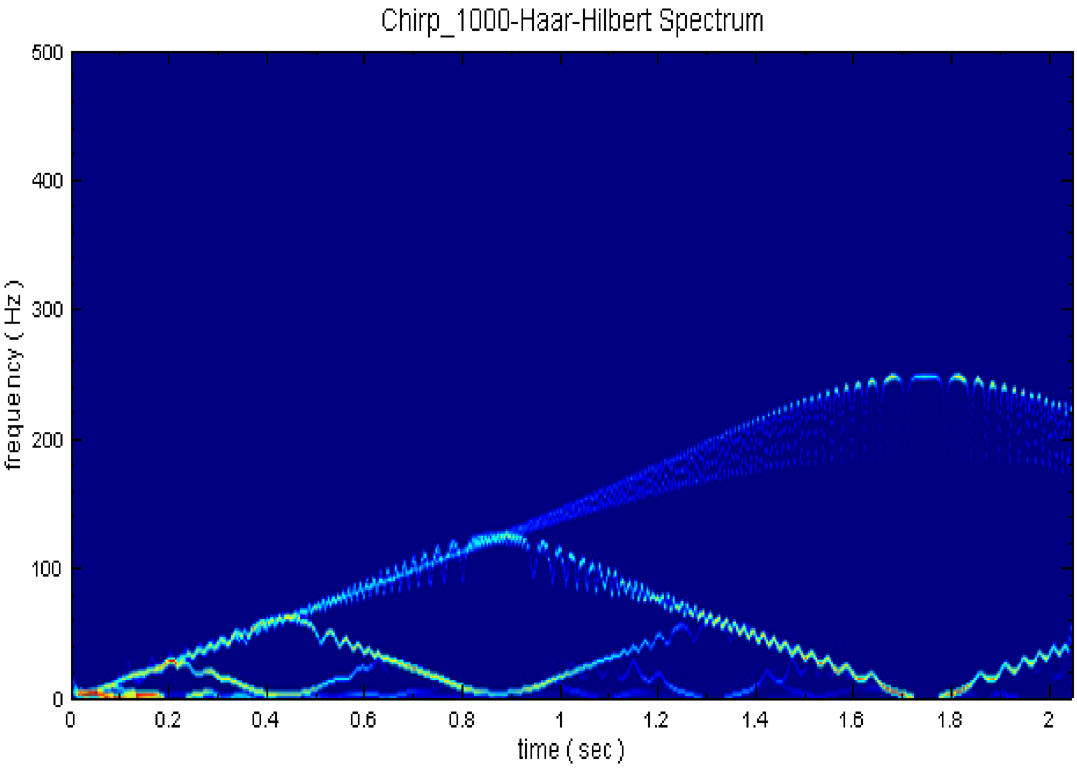
Hilbert Spectrum can also show the instaneous frequency from Haar Wavelet Transform. Connect Chirp_1000 to Compute / Transform / Haar Wavelet Transform, and set Properties / FilterBank levels to 9, set Resampling method to Spline. Then follow steps 2~3 to do time-frequency calculation.






Chirp_1000-Haar
12
10
8
6
4
2
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
time (sec)


chirp1000
Viewer [1]
RCADA::EEMD
Viewer2 [2]
HSPEC
TF Viewer [3]
Haar
Viewer3 [4]
HSPEC2
TF Viewer2 [5]



Related Functions

RADAC EEMD, IMF Properties, TFA Viewer, Haar Transform.

References

Proc. R. Soc. Land A 1998(903-995).

The Hilbert-huang Transform And Its Applications Huang, by Norden E. (EDT)/Shen, Samuel S. (EDT) World Scientific Pub Co Inc.

3.5.5 Marginal Time/Marginal Frequency

Introduction

Marginal Time is to obtain a distribution in time domain by integrating a signal, $x(t,\omega)$ with respect to the frequency, where the signal has been processed by time-frequency analysis. Mathematically, this operation can be written as

$$x(t) = \int_{-\infty}^{\infty} X(t,\omega)\,d\omega$$

where $X(t,\omega)$ is a two-dimensional time-frequency array, $x(t)$ is a time domain distribution, i.e., Marginal Time.

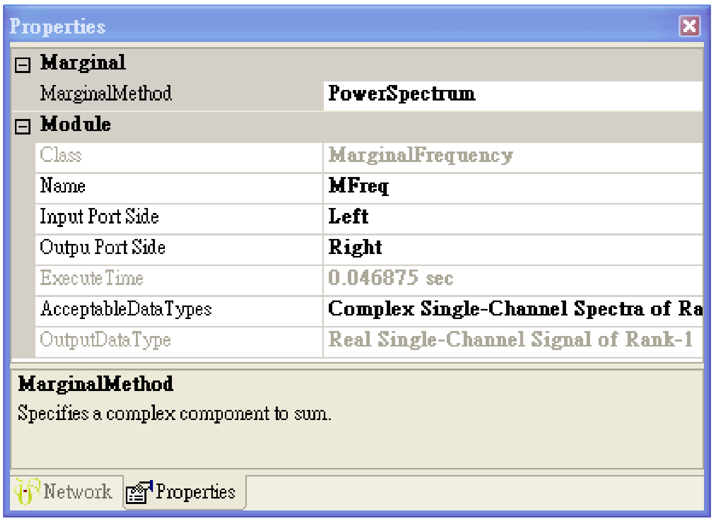
If integration is performed with respect to the time, frequency distribution of Marginal Frequency could be obtained. This operation can be written as

$$x(\omega) = \int_{-\infty}^{\infty} X(t,\omega)\,dt$$

Properties

This module accepts input of Spectra (which could be real number or complex number, single channel, Regular). The output format is real, single channel, and Regular signal.

The property is *Marginal Method* which mainly is used to process two-dimensional time-frequency complex array. Different from time/frequency integration, 7 options are provided as listed in the tables below.

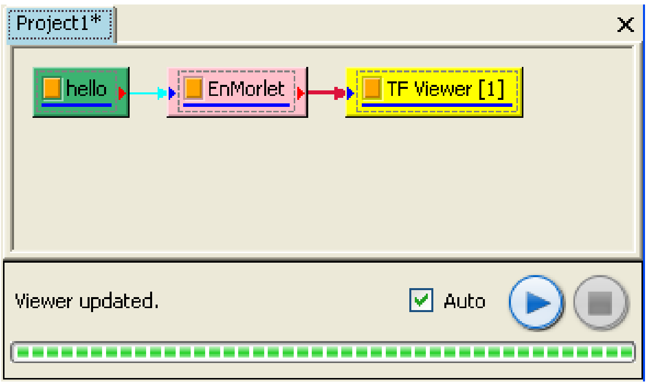


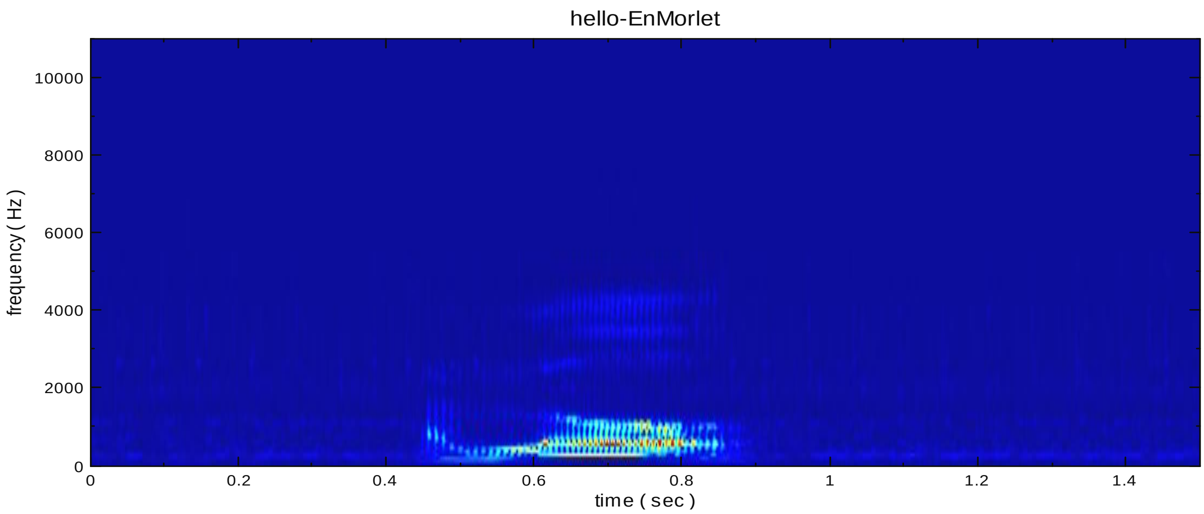
Option	Definition
Complex	Integrate with respect to real part and imaginary part. $\mathrm{Re}(x(t)) = \int \mathrm{Re}(X(t,\omega))d\omega$, $\mathrm{Im}(x(t)) = \int \mathrm{Im}(X(t,\omega))d\omega$ $\mathrm{Re}(x(\omega)) = \int \mathrm{Re}(X(t,\omega))dt$, $\mathrm{Im}(x(\omega)) = \int \mathrm{Im}(X(t,\omega))dt$
Magnitude	Integrate after performing norm operation on time-frequency signal. $x(t) = \int \sqrt{\mathrm{Re}(t,\omega)^2 + \mathrm{Im}(t,\omega)^2}\, d\omega$ $x(\omega) = \int \sqrt{\mathrm{Re}(t,\omega)^2 + \mathrm{Im}(t,\omega)^2}\, dt$
Phase	Integrate after calculating the phase angle $x(t) = \int \tan^{-1}\left(\frac{\mathrm{Im}(t,\omega)}{\mathrm{Re}(t,\omega)}\right)d\omega$ $x(\omega) = \int \tan^{-1}\left(\frac{\mathrm{Im}(t,\omega)}{\mathrm{Re}(t,\omega)}\right)dt$
RealPart	Integrate the time-frequency signal with respect to the real part $x(t) = \int \mathrm{Re}(t,\omega)d\omega$ $x(\omega) = \int \mathrm{Re}(t,\omega)dt$
ImagPart	Integrate the time-frequency signal with respect to the imaginary part $x(t) = \int \mathrm{Im}(t,\omega)d\omega$ $x(\omega) = \int \mathrm{Im}(t,\omega)dt$
PowerSpectrum	Integrate after performing norm^2 operation on time-frequency signal. $x(t) = \int [\mathrm{Re}(t,\omega)^2 + \mathrm{Im}(t,\omega)^2]d\omega$ $x(\omega) = \int [\mathrm{Re}(t,\omega)^2 + \mathrm{Im}(t,\omega)^2]dt$

Example

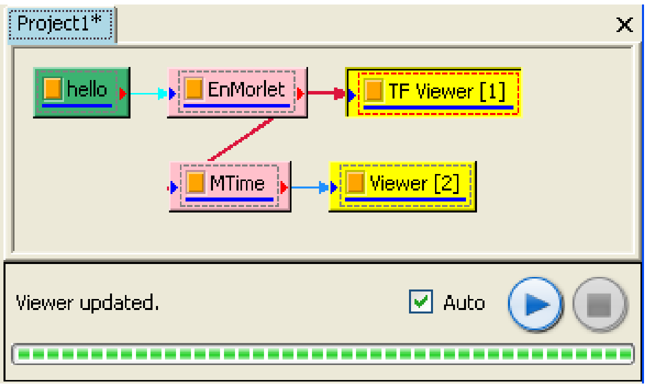
In this example, use audio signal, hello.wav, as a signal source, perform Enhanced Morlet Transform in DataDemon to perform time-frequency analysis, and then use *Marginal Time* to calculate the frequency distribution in time domain.

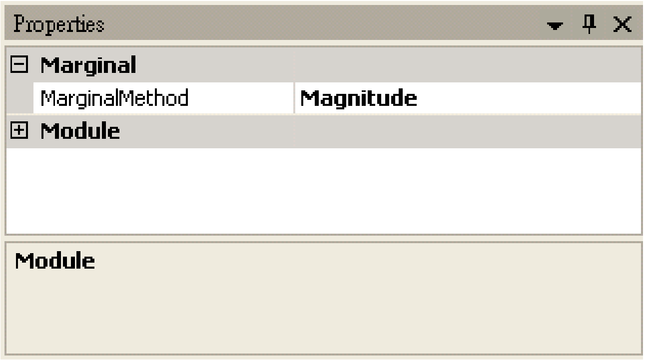
1. Press the in the Network tools, or use *Source→Import data from file* to read the signal file, hello.wav, in the installation directory (default to be C:\Program Files\DynaDx\DataDemon\data). Next, perform *Compute→TFA→Enhanced Morlet Transform* and then use *Viewer→Time Frequency Viewer* to plot the result.

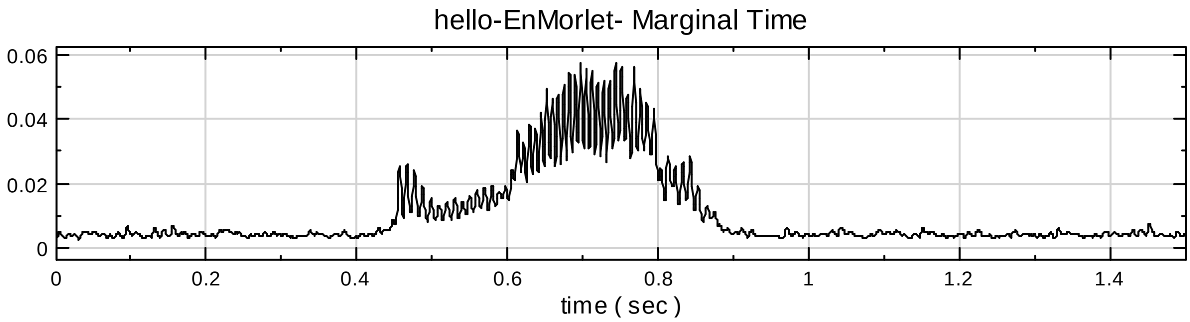




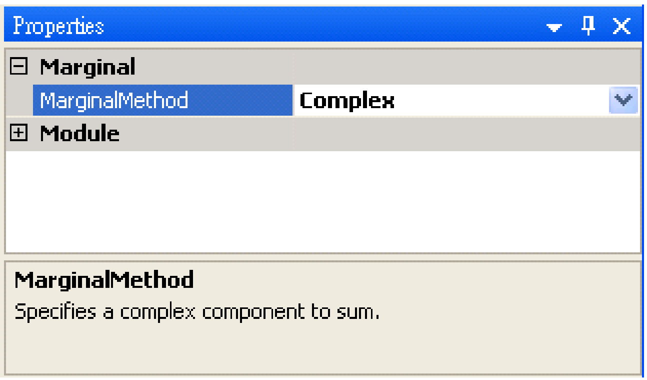
2. After EnMorlet, perform *Compute→TFA→Marginal Time*, whose property of The MarginalMethodg is set as Magnitude. In the result, the x-axis is set as time while the y-axis is set as amplitude.

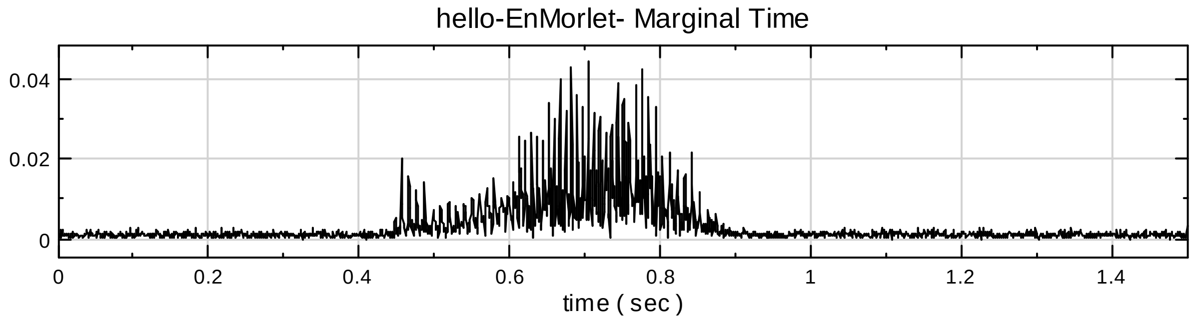




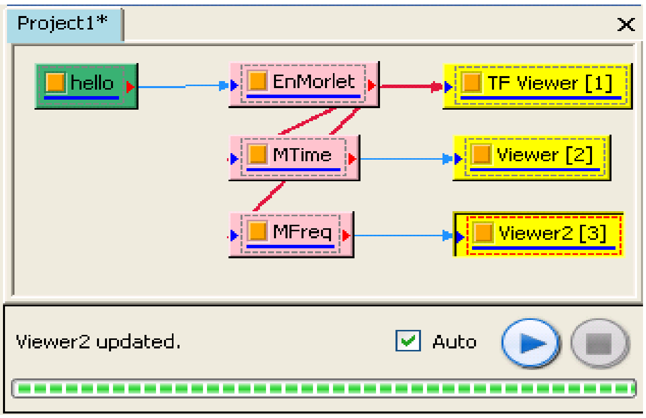


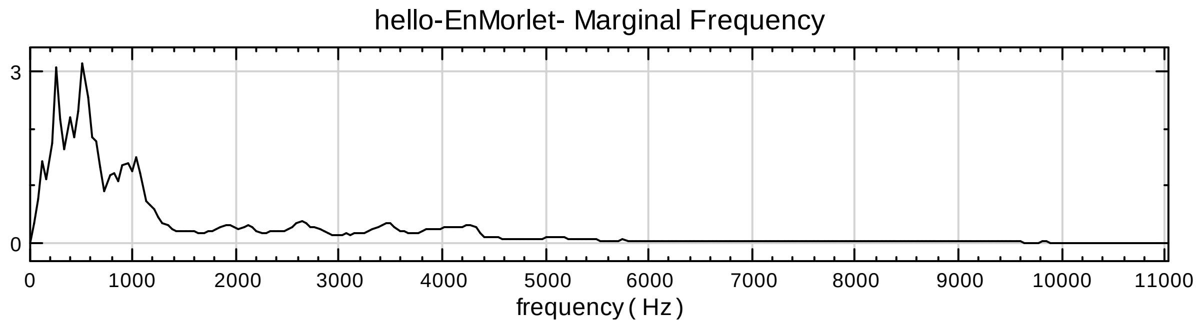
3. Change MarginalMethod to Complex, which means that Marginal Time integrates the time-frequency signal directly.





4. Next, perform *Compute→TFA→Marginal Frequency* following EnMorlet. In the result, the x-axis is frequency while the y-axis is amplitude.





Related Functions

STFT, Morlet Transform, Enhanced Morlet Transform.

3.6 Transform

This module provides common transforms for different signal processing.

1. Fourier Transform/Inverse Fourier Transform
2. Discrete Cosine Transform/Inverse Discrete Cosine Transform
3. Haar Wavelet Transform
4. Hilbert Transform/Inverse Hilbert Transform
5. Auto Correlation: Calculate the auto correlation between signals
6. Cross Correlation: Calculate the cross correlation between signals
7. Multi-Scale Entropy(MSE): Calculate signal multi-scale entropy

3.6.1 Fourier Transform and Inverse Fourier Transform

Fourier Transform converts a time signal to a frequency signal for checking the frequency and amplitude distribution in the signal. The frequency signal could be converted back to time signal by Inverse Fourier Transform. This method is widely used in communication, voice signal, system analysis, and other scientific fields.

Introduction

Let $X = \{x_0\ \ x_1\ \ x_2 \cdots x_{N-1}\}$ be a N-length time signal, x_n be the n^th signal, $0 \le n \le N-1$, the discrete Fourier Transform of signal X is defined as a N-length series,

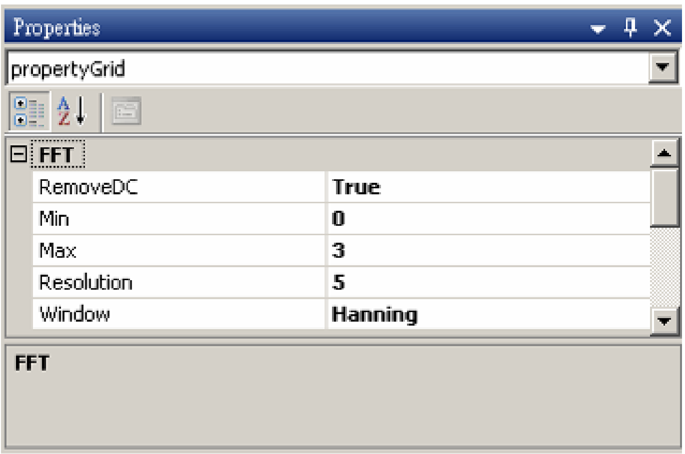
$$F(x_k) = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N}kn}\ ,\ 0 \le k \le N-1$$

The Inverse Fourier Transform is defined as follows,

$$x_n = N\sum_{k=0}^{N-1} X_k e^{i\frac{2\pi}{N}kn}\ ,\ 0 \le n \le N-1$$

Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular). The definition of properties and corresponding setting are given below.

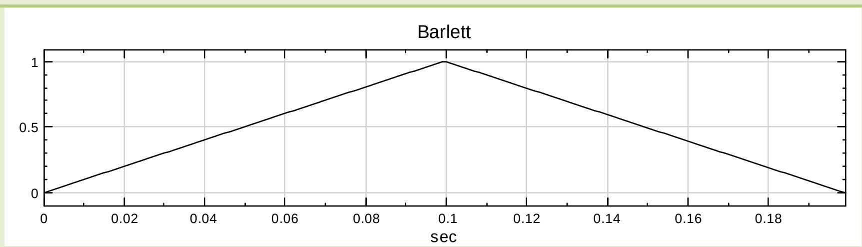
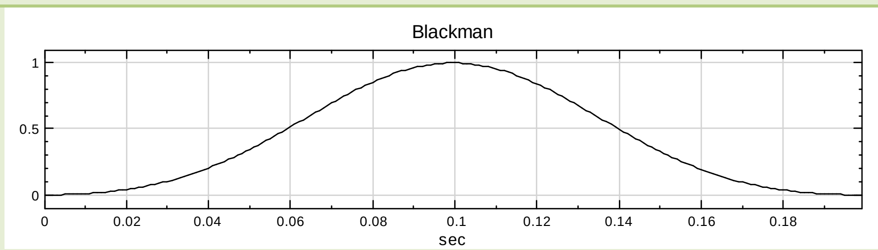
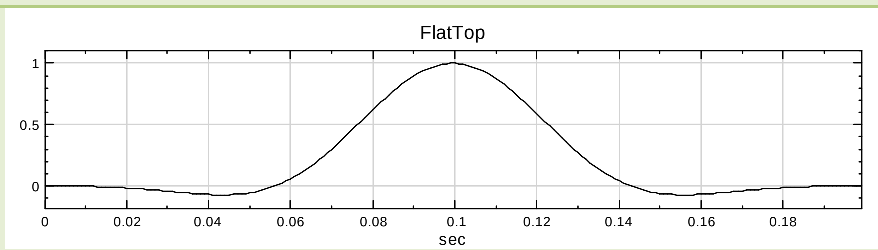


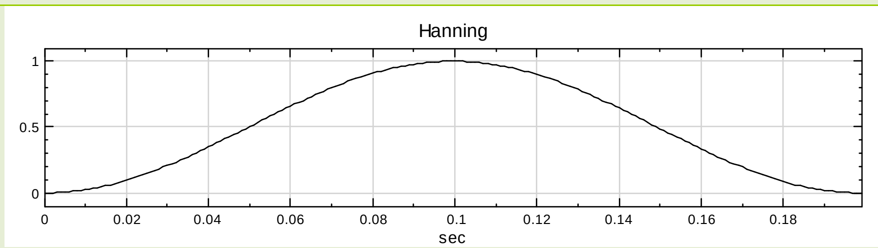
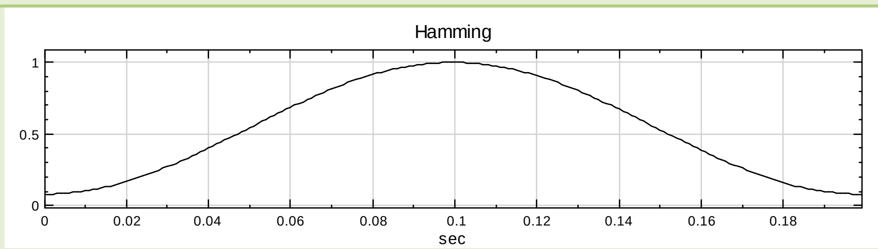
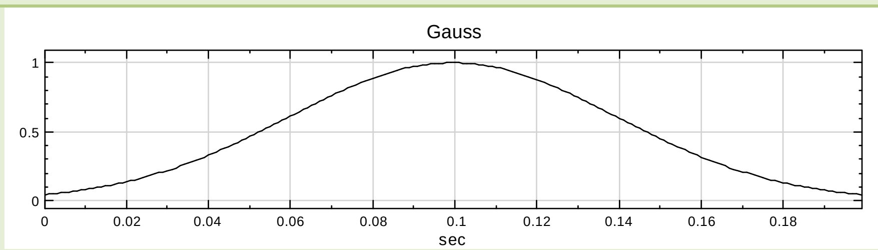
The property of RemoveDC is used to remove the average of signal. The properties of Min and Max define the frequency range of FFT. The Property of

Resolution is used to duplicate the signal to double the signal length, k, of Fourier Transform, for better spectrum resolution. In the Window properties, there are 6 common window functions which can be used to smooth the discrete signal and therefore remedy the numerical error caused by boundary effects.

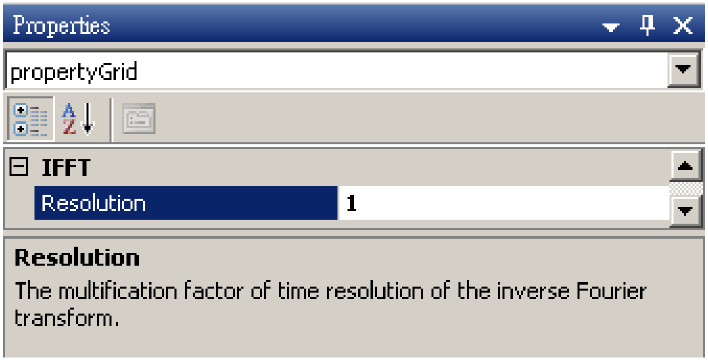
Property Name	Property Definition	Default Value
RemoveDC	To remove the shift along the y-axis, making the signal average to be zero.	True
Min	To set the lower frequency boundary of the Fourier Transform	0
Max	To set the upper frequency boundary of the Fourier Transform, which varies based on the input signal	auto
Resolution	To adjust the Fourier Transform resolution. The approach is to multiply the input data point with the Resolution for increasing the transform resolution, then use Cherp Z Transform to obtain high-resolution Fourier Transform.	1
Window	Use window function to reduce the leakage effect on the Transform. The window functions include 6 types: Barlett, Blackman, Flat Top, Hanning, Hamming, and Gauss, whose definitions are given below.	none

Window Function

Window Function	Definition and Diagram
Barlett	 $w[n]=\begin{cases} \frac{2n}{N-1}, & 0\le n\le \frac{N-1}{2} \\ 2-\frac{2n}{N-1}, & \frac{N-1}{2}\le n\le N-1 \\ 0, & otherwise \end{cases}$
Blackman	 $w[n]=\begin{cases} \frac{1-\alpha}{2}-\frac{1}{2}\cos(\frac{2\pi}{N-1}n)+\frac{\alpha}{2}\cos(\frac{4\pi}{N-1}n), & 0\le n\le N-1 \\ 0, & otherwise \end{cases}$
FlatTop	 $w[n]=\begin{cases} 1-1.93\cos\left(\frac{2\pi n}{N-1}\right)+1.29\cos\left(\frac{4\pi n}{N-1}\right)-0.388\cos\left(\frac{6\pi n}{N-1}\right)+0.032\cos\left(\frac{8\pi n}{N-1}\right), \\ 0\le n\le N-1 \\ 0, \quad otherwise \end{cases}$

Window Function	**Definition and Diagram**
Hanning	 $w[n]=\begin{cases}0.5-0.5\cos(\frac{2\pi\cdot n}{N-1}), & 0\le n\le N-1\\ 0, & otherwise\end{cases}$
Hamming	 $w[n]=\begin{cases}0.53836-0.46164\cos(\frac{2\pi\cdot n}{N-1}), & 0\le n\le N-1\\ 0, & otherwise\end{cases}$
Gauss	 $w[n]=e^{-\frac{1}{2}\left(\frac{n-\frac{N-1}{2}}{\sigma\frac{N-1}{2}}\right)^2}, \sigma\le 0.5$

The properties of Inverse Fourier Transform are given as follows.

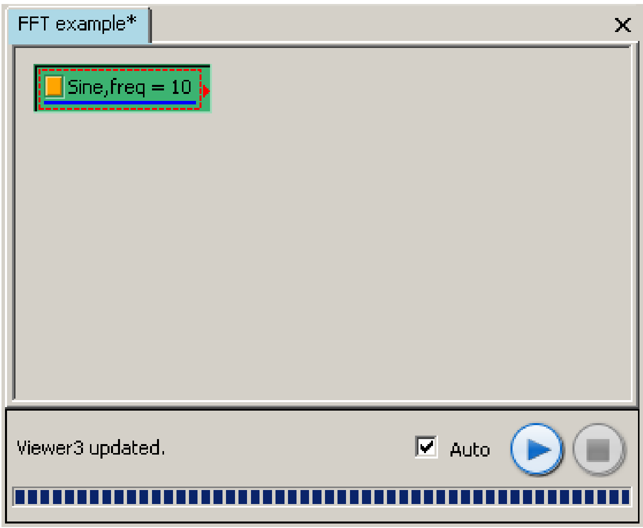


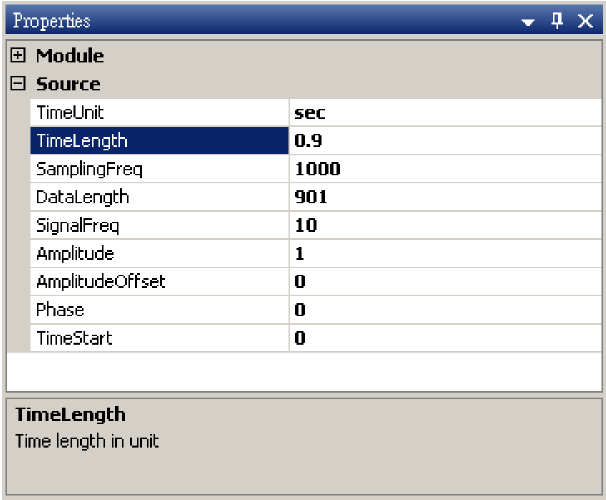
The property of Inverse Fourier Transform is Resolution, which has identical meaning as that in Fourier Transform. The number of signals in Inverse Fourier Transform would be twice as many as the Resolution value.

Example

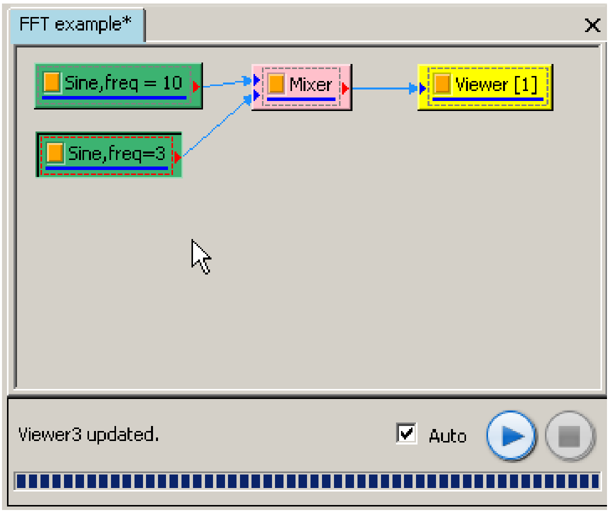
This example uses Source Module to generate a combined signal of two sine waves.

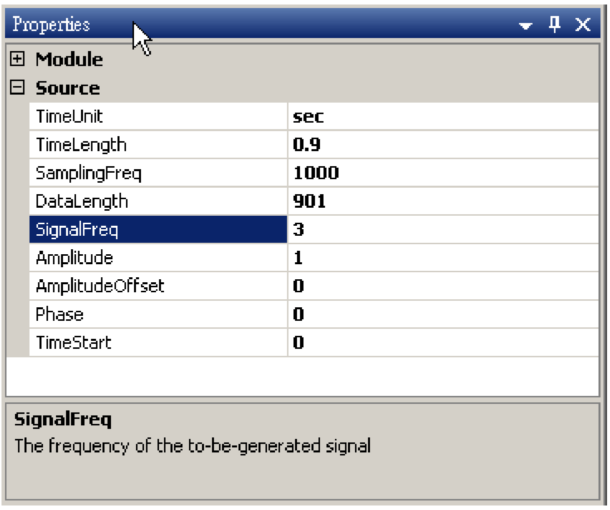
1 In the Network window, use Source→Sine to create a sine wave. In window of Properties, change the Name filed to Sine, freq =10. The default value of Signal frequency is 10 Hz. Change TimeLength field to 0.9 sec.

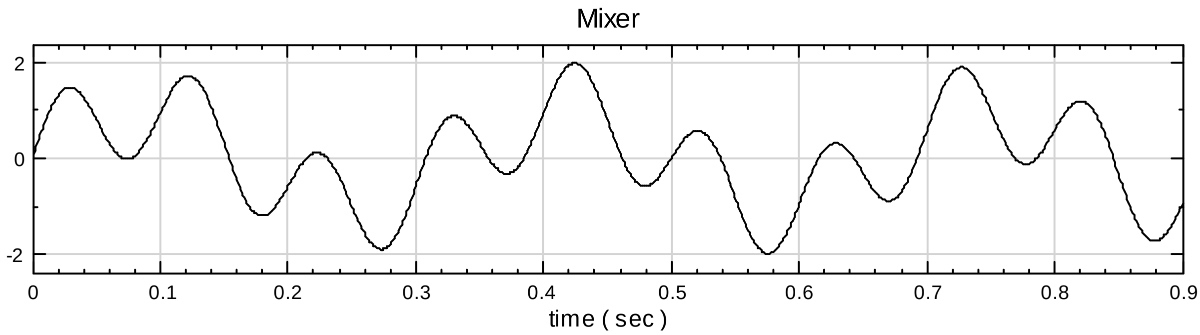




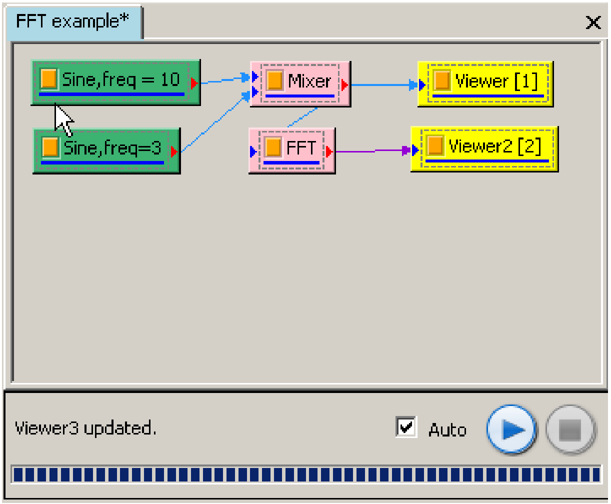
2 Create another sine wave and set it Siganl Frequency to 3, TimeLength to 0.9 second. Then, add Compute→Mathematics→ Mixer modules to combine these two signals and use View→Channel Viewer to plot the output.

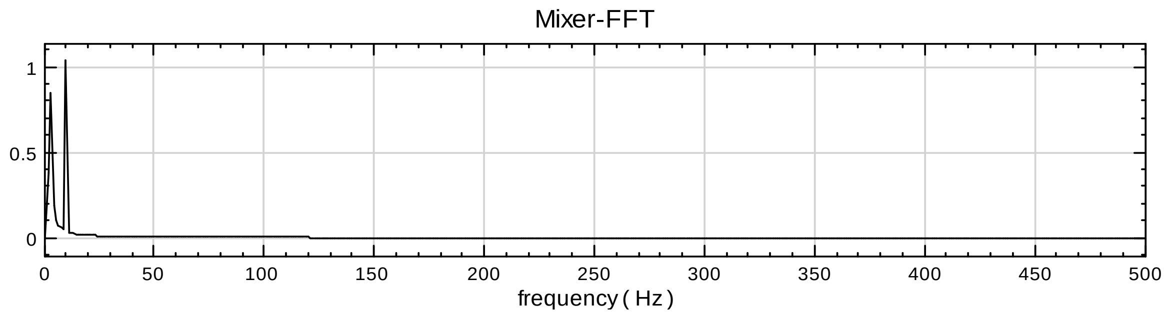




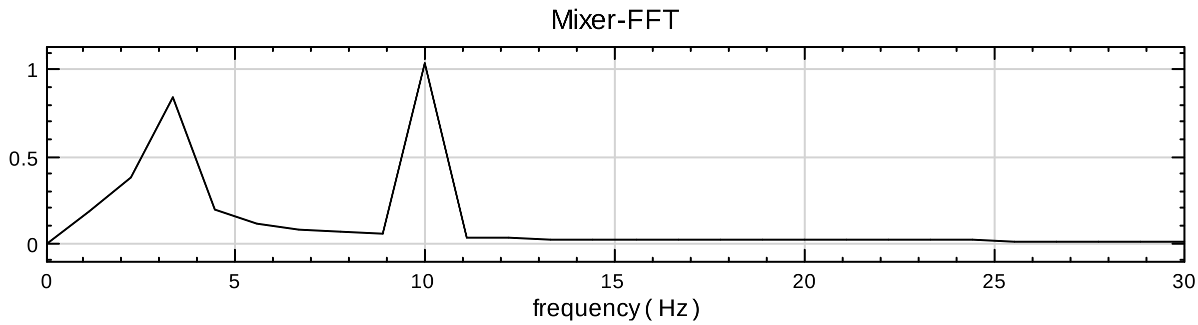


3 On the icon of Mixer, select Compute→Transform→Fourier Transform to perform FFT and then use Channel Viewer to plot the spectrum in the left window.



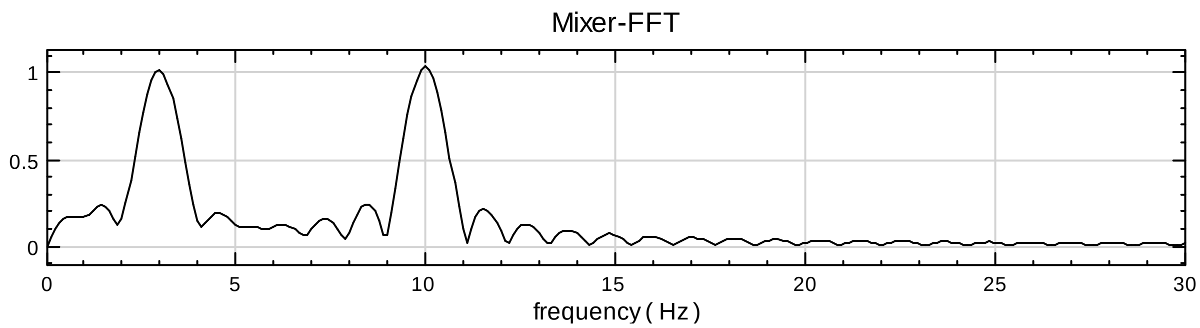


Because most frequencies are less than 20Hz and the default xmax is 500 in Properties of Viewer, set this field to 30 for better observation.



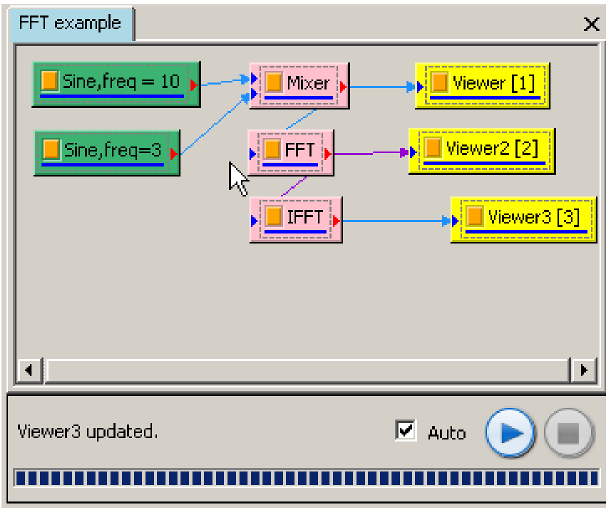
4 In the spectrum diagram generate by FFT, frequency components mainly concentrate at 10Hz and 3 Hz. However, the magnitude around 3Hz is under-estimated due to the low frenquency. This could be enhanced by changing the resolution. Click on the FFT icon, change the Properties/Resolution to 5 to obtain new result.

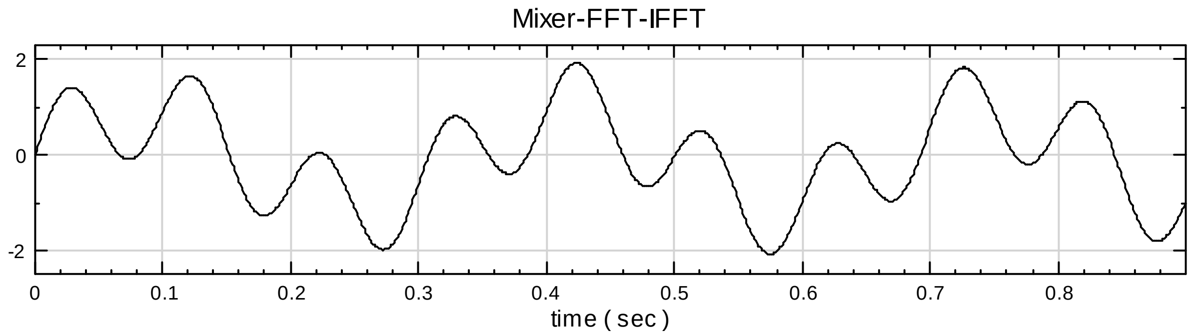
It is shown that the spectrum around 3 Hz has been improved significantly after changing the resolution. Please note that increasing the resolution would result in multiplication of output data length in FFT. In this example, the input data length is 901, the output length of FFT is 451 when the resolution is 1. After changing



the resolution to 5, the output length would increase 5 times to 2255.

5 Change the FFT resolution back to 1, right click the FFT icon to select Compute→Transform→Inverse Fourier Transform, then use *Channel Viewer* to view the result. It is the same as the original signal.





Related Functions

Short-Term Fourier Transform

Reference

http://en.wikipedia.org/wiki/Fourier_transform

3.6.2 Discrete Cosine Transform and Inverse Cosine Transform

Discrete Cosine Transform（DCT）converts signals to a series of cosine, which is similar to the real part in the FFT. After the transformation, signal energy mostly concentrates in low frequency area. This method is applicable in audio and image compression, numeric solution of partial differential equation and other fields.

Introduction

There are approximately 8 types of common DCT. This module uses the 2nd type. Let $X = \{x_0, x_1 ... x_{N-1}\}$ be a N-length time series, the DCT is defined as below,

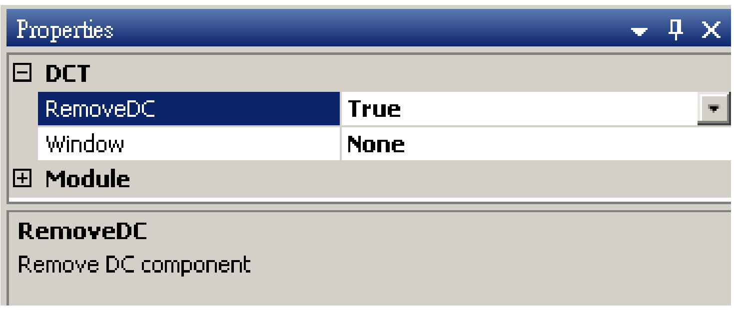
$$Y_k = C_k \sum_{n=0}^{N-1} X_n \cos\frac{(2n+1)\pi k}{2N}, \quad \begin{cases} C_k = \dfrac{1}{(N)^{\frac{1}{2}}}, \text{ for } k = 0 \\ C_k = (\dfrac{2}{N})^{\frac{1}{2}}, \text{ for } k > 0 \end{cases}$$

where C_k is the DCT coefficient of DCT. The inverse DCT is defined as below.

$$X_n = C_n \sum_{k=0}^{N-1} Y_k \cos\frac{(2n+1)\pi k}{2N}, \quad \begin{cases} C_n = \dfrac{1}{(N)^{\frac{1}{2}}}, \text{ for } k = 0 \\ C_n = (\dfrac{2}{N})^{\frac{1}{2}}, \text{ for } k > 0 \end{cases}$$

Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular). The output formats are real number, signal channel, Regular signal. The properties and settings of the DCT are introduced below. The inverse DCT does not have this type of property.

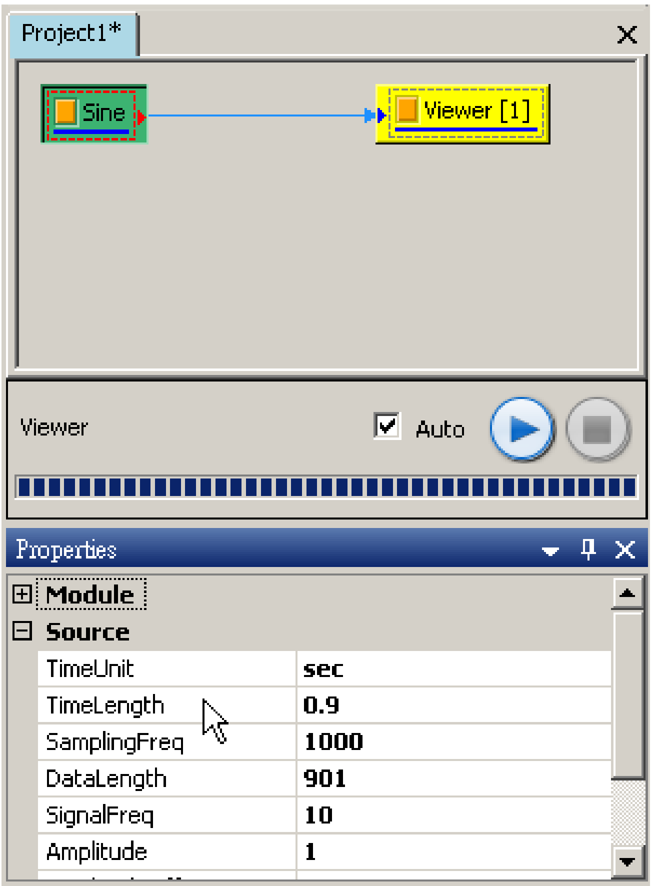


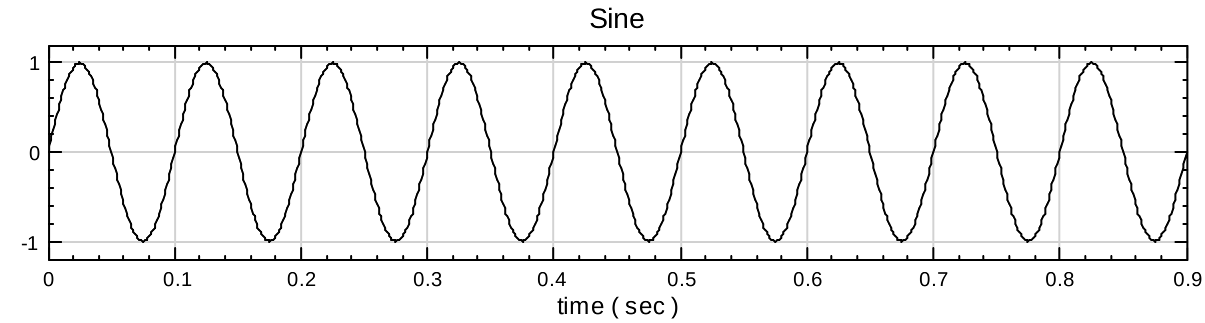
Property Name	Property Definition	Default Value
RemoveDC	To remove the signal shift in the amplitude.	True
Window	To decide whether window filter are needed before FFT processing. Please reference to FFT for details	None

Example

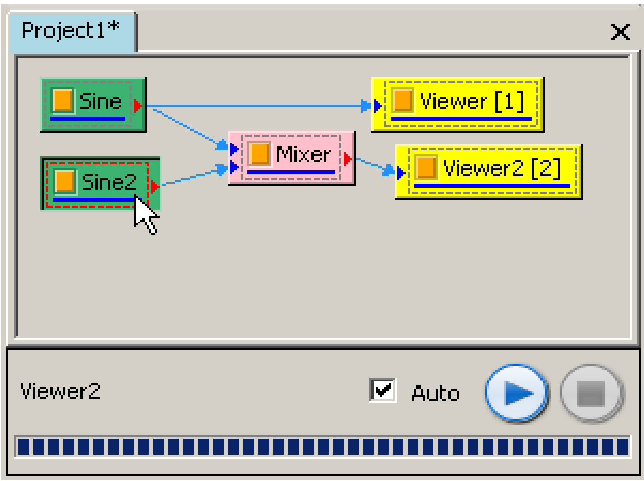
In this example, use Source to create a combined signal of two wave signals. The 1st signal has frequency of 10Hz, TimeLength of 0.9 second. The 2nd signal has frequency of 3Hz and TimeLength of 0.9 second. Then use DCT to calculate the signal spectrum and use Inverse DCT on this spectrum to obtain the original signal.

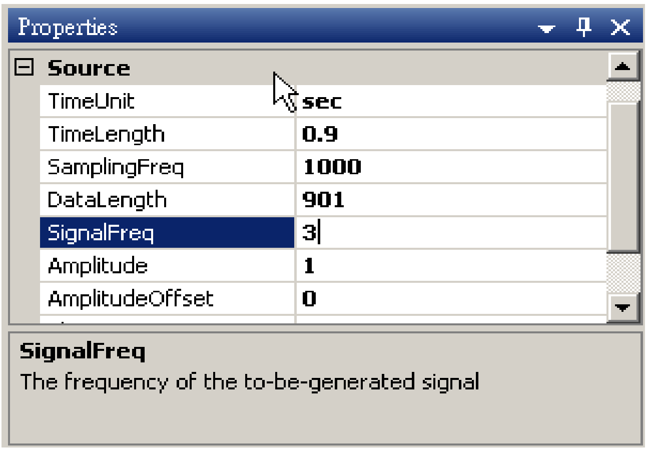
1 In the Network window, select Source→Sine to create a sine wave. Change its TimeLength field to 0.9 seccond.

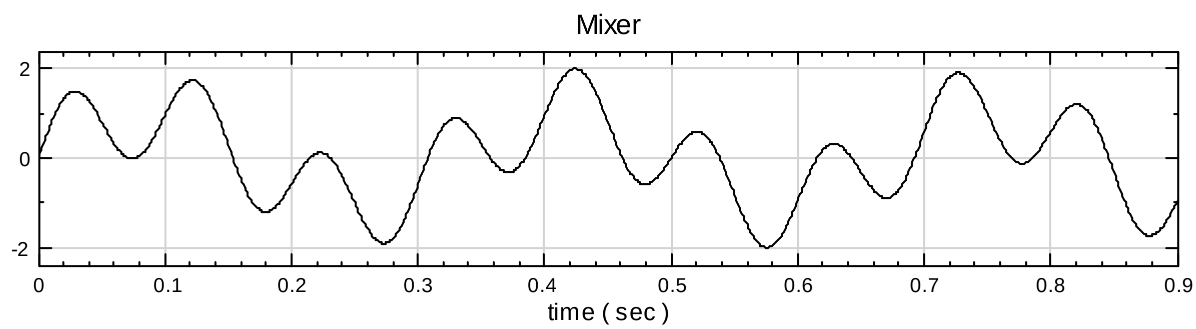




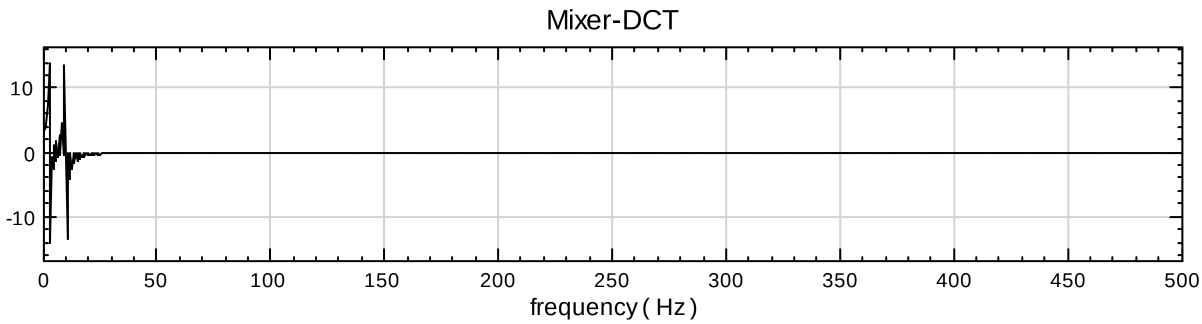
2 Similar to the step1, create another sine wave which has frequency of 3 Hz and TimeLength of 0.9 second. Add a *Compute→Mathematics→ Mixer* component to mix these two signals and use *Viewer→Channel Viewer* to plot the output.

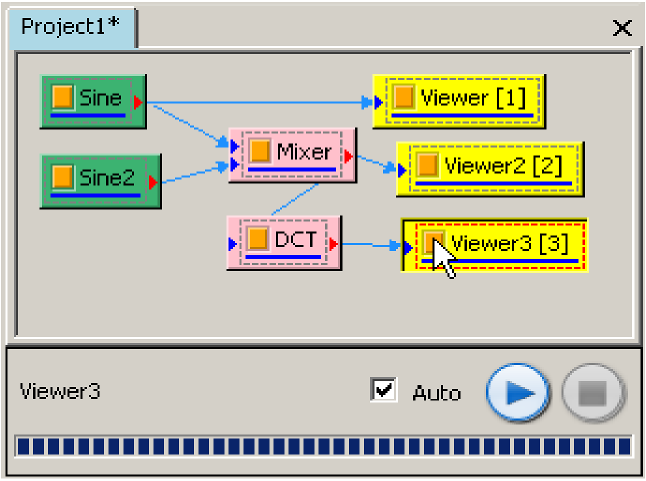




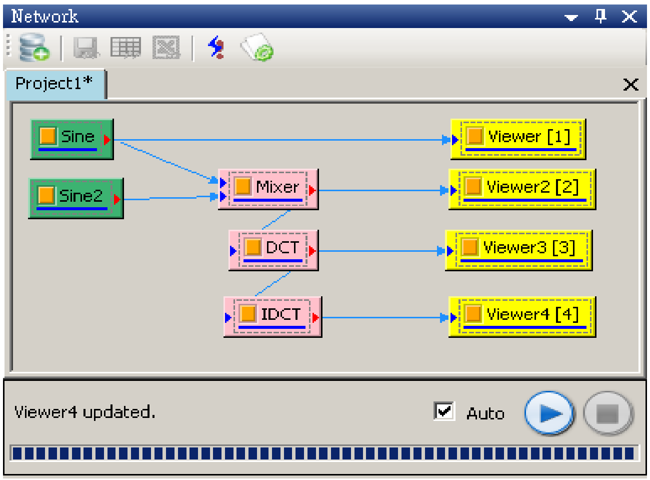


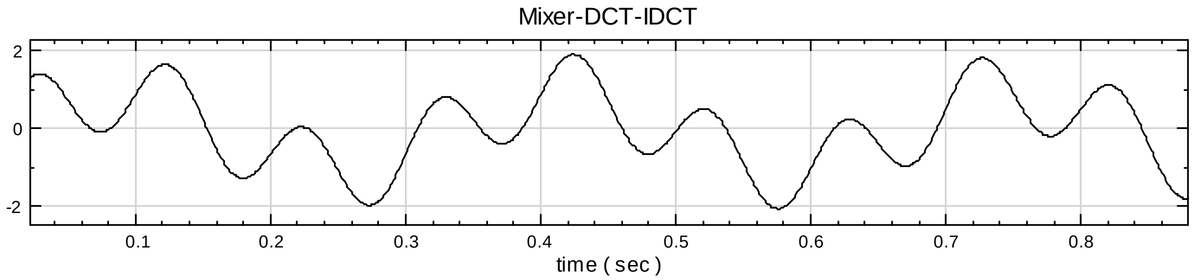
3 Perform *Compute→Transform→Discrete Cosine Transform* on the output generated by *Mixer* and plot the result using Channle Viewer. It is shown that the signal spectrum mostly concentrates in the low frequency range.





4 Final, perform *Compute→Transform→Inverse Cosine Transform* to convert the signal back to the original wave.





Related Functions

Fourier Transform, Viewer.

Reference

http://en.wikipedia.org/wiki/Discrete_cosine_transform

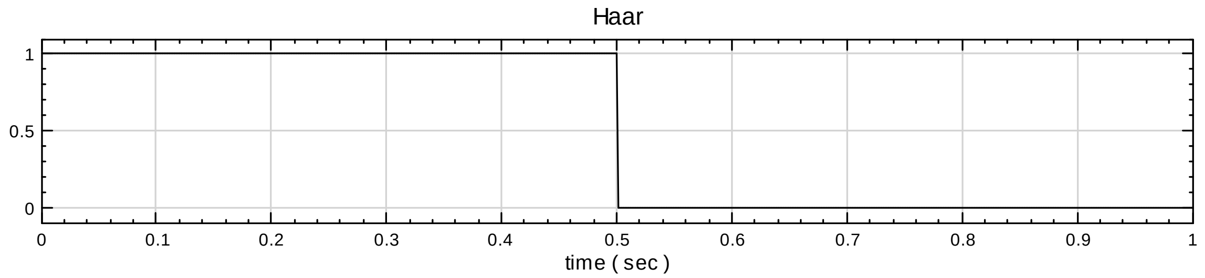
3.6.3 Haar Wavelet Transform

Haar Wavelet is the first published Wavelet function, proposed by Alfréd Haar. Harr wavelet is the simplest orthogonal wavelet which is the fundament of binary wavelet transfrom. However, because it is not a continuous function, its performance is not the best as a fundamental wavelet.

Introduction

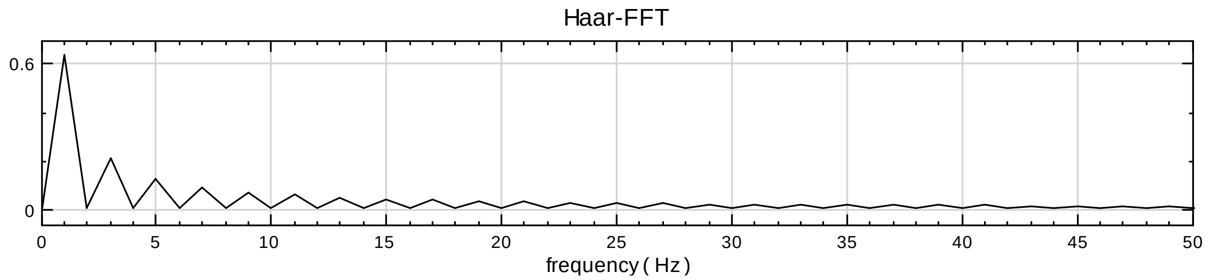
The mother wavelet of Haar Wavelet can be denoted as follows.

$$\psi(t)=\begin{cases}1, 0\le t\le \frac{1}{2}\\ -1, \frac{1}{2}\le t\le 1\\ 0, otherwise\end{cases}$$



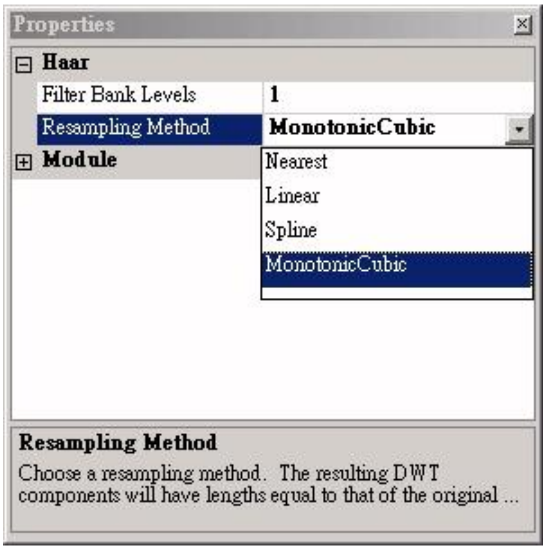
The FFT of Haar is

$$\phi(\omega)=\frac{1-2e^{-i\frac{\omega}{2}}+e^{-i\omega}}{i\omega}$$



Properties

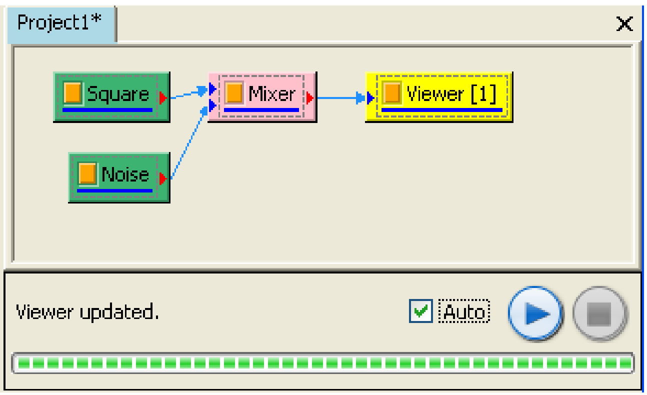
This module accepts input of Signal (which could be real number, single channel, Regular) and Audio (which could be real number, single channel, Regular). Detailed properties are given as follows.

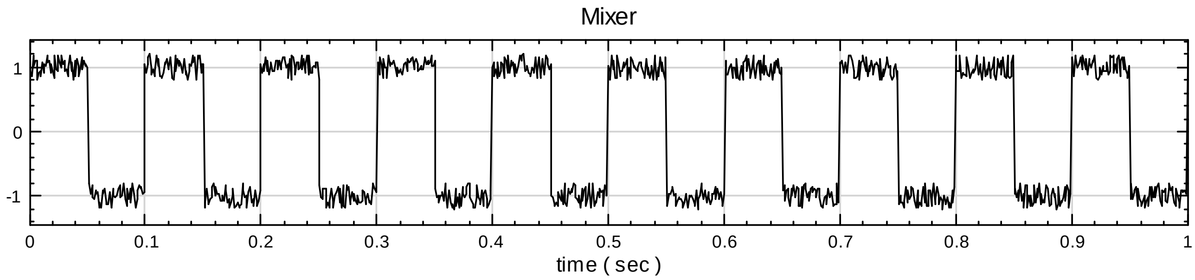


Property Name	Property Definition	Default Value
Filter Bank Level	The number of orthogonal basis starting from 0. 1 means 2 basis, 2 means 3 basis, and so on.	1
Resampling Method	Resampling Approximation method. Please reference to Resampling	Linear

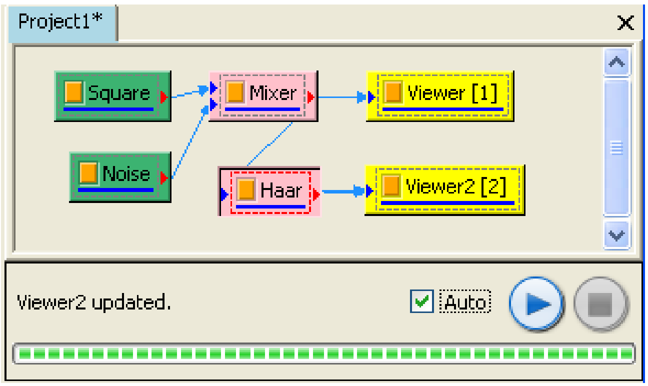
Example

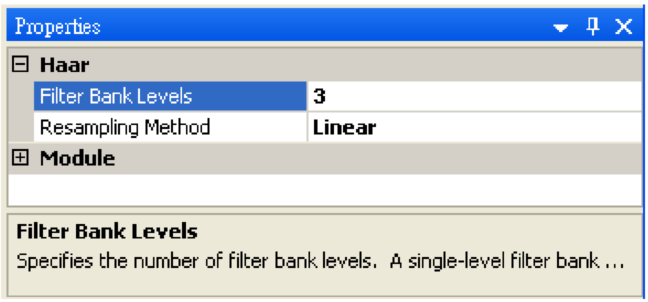
1. Using all default values to create a square wave, and then use *Math→Mixer* to mix one white noise which has amplitude of 0.2. Finally, use *Viewer→Channel Viewer* to plot the result.



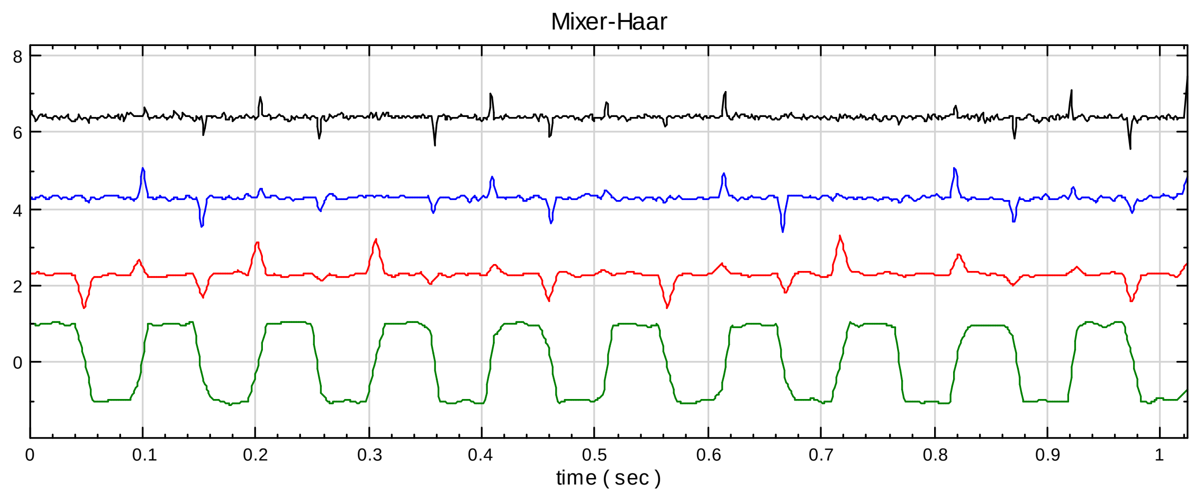


2. Connect the mixed signal with *Compute→Transform→Haar transform* and set the *Filter Bank levels* to 3, and then connect *Channel Viewer* to plot the calculation result.



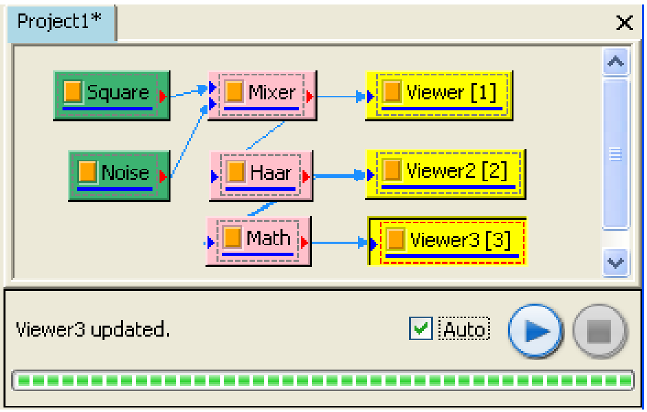


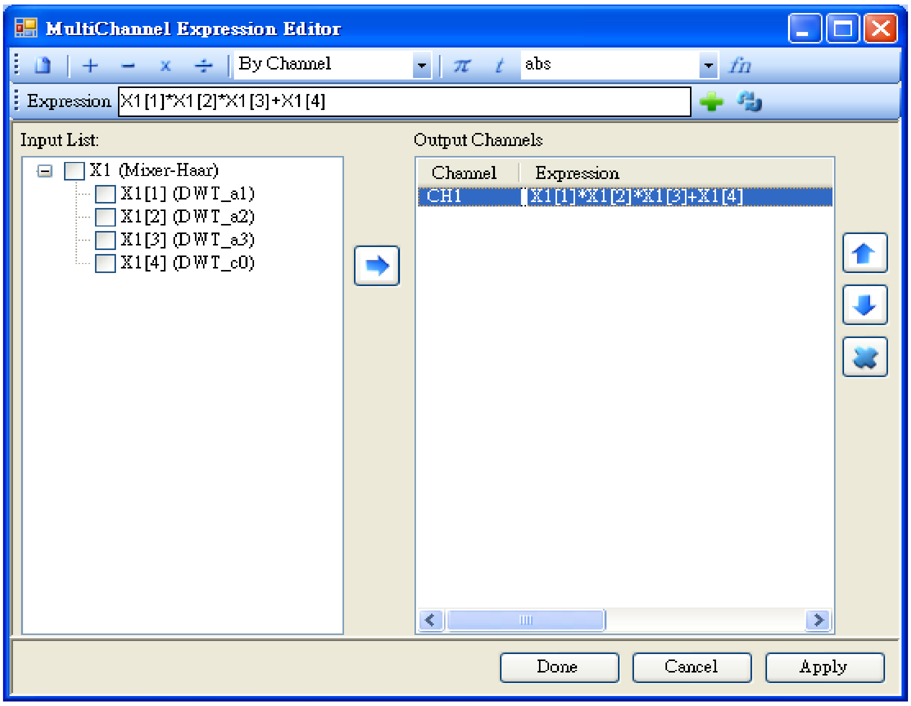
Change the Viewer Height in Viewer[2] to 300 for better observation.

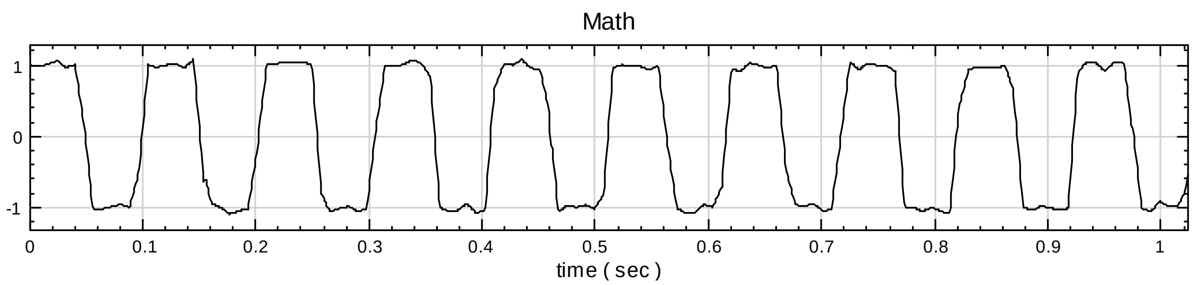


Obviously, the characteristic of square wave is mainly shown in the 4th curve. At the discontinous points of the original signal, the jump characteristic is also perserved in the 2nd and 3rd curves significantly. In addition, noises are mainly shown in the 1st ~ 3rd curves.

3. To preserve the square wave characteristics in the 1st ~ 3rd curves and reduce the effects caused by noises, use Mathematics→Math to multiply the 1st, 2nd, and 3rd signals and then plus the 4th signal. The result is shown below.

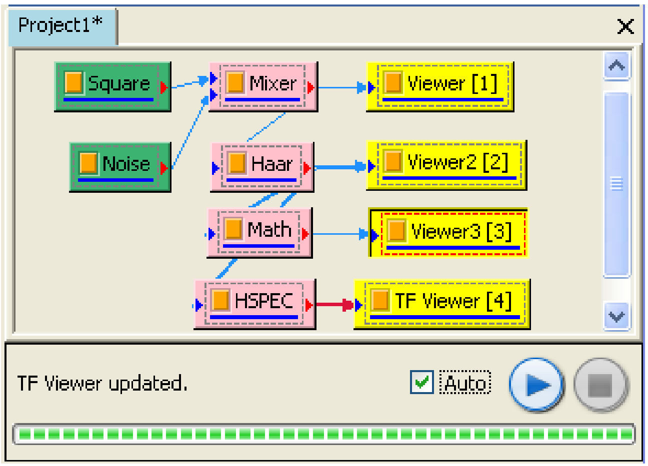


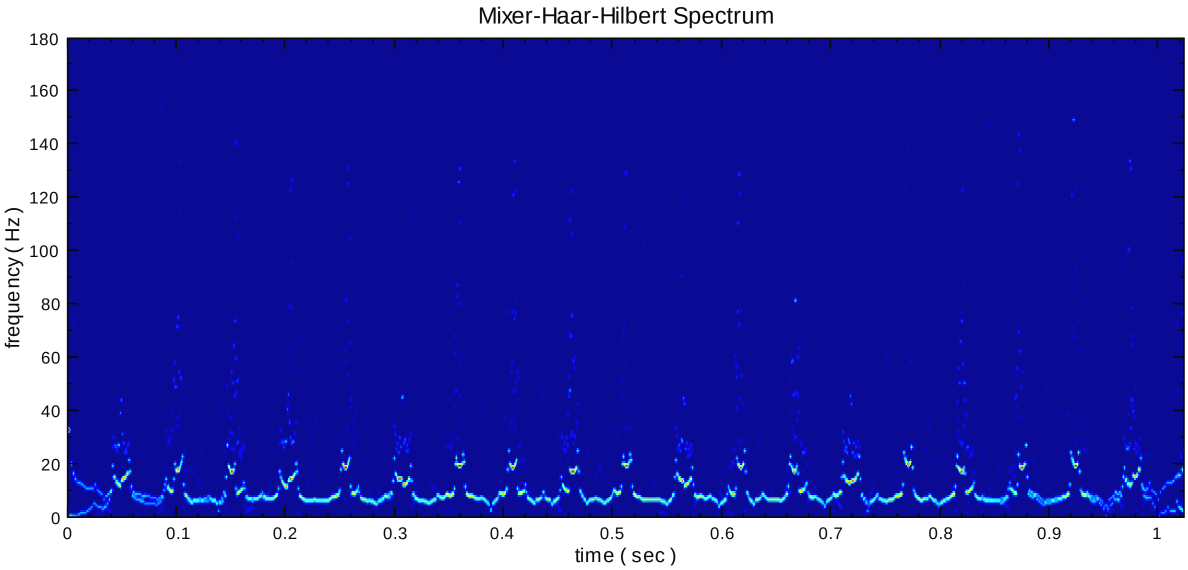




Processed in this way, the visible noises have been eliminated while the characteristics of the original square wave is preserved.

4. Similar to the EEMD, the result of Haar Transform could be processed by Hilbert Spectrum to obtain spectra diagram.





Related Functions

Mixer, Multiplier, Fourier Transform, Hilbert Spectrum.

Reference

http://amath.colorado.edu/courses/4720/2000Spr/Labs/Haar/haar.html

3.6.4 Hilbert Transform

Hilbert Transform executes convolution on an input signal $x(t)$ and $\frac{1}{\pi t}$, i.e., $\int_{-\infty}^{\infty}\frac{1}{\pi}\frac{X(\tau)}{t-\tau}d\tau$, to convert a real time signal to an analytic complex signal, whose real part is the input signal while the imaginary part is the result of convolution on the original signal. Based on the analytic complex signal, instantaneous frequency, instantaneous amplitude can be defined. Hilbert Transform has shown good performance in communication system and wireless signal processing and analysis.

Introduction

Let $X(t)$ be a time series, its Hilbert Transform $y(t)$ can be defined as

$$Y(t)=\frac{1}{\pi}P.V.\int_{-\infty}^{\infty}\frac{X(\tau)}{t-\tau}d\tau$$

and an analytic function $Z(t)$ is defined as

$$Z(t)=X(t)+iY(t)=a(t)e^{i\theta(t)},\ a(t)=\left[X^{2}(t)+Y^{2}(t)\right]^{\frac{1}{2}},\ \theta(t)=\arctan\left(\frac{Y(t)}{X(t)}\right)$$

where $a(t)$ is the amplitude, i.e., the envelop of the original signal, while $\theta(t)$ is the phase angle. In the Polar Coordinate representation of the analytic function $Z(t)$, the instantaneous frequency, ω, of the Hilbert Transform could be defined as below,

$$\omega(t)=\frac{d\theta(t)}{dt}$$

For more details, please reference to Hilbert Spectrum.

Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular). The output format is complex, single channel, Regular signal.

Theres are two properties in Hilbert transform. The Output Type is used to set the signal output of conversion. Available options are Complex, Split Complex, Unwrapped Phase, InstantFrequency and Instant Amplitude. The calculation methods

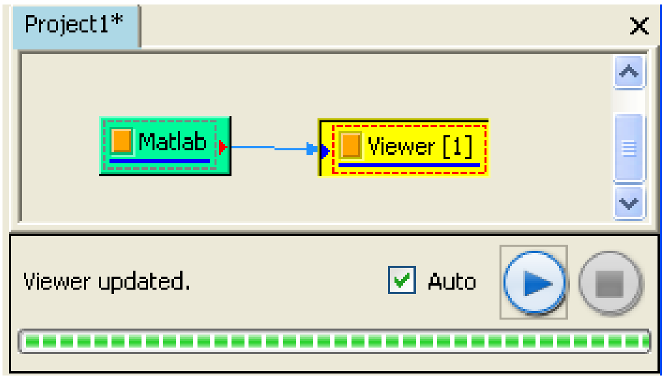
of InstantFrequency include Simple and Barne. The details of these options are given in the tables below.

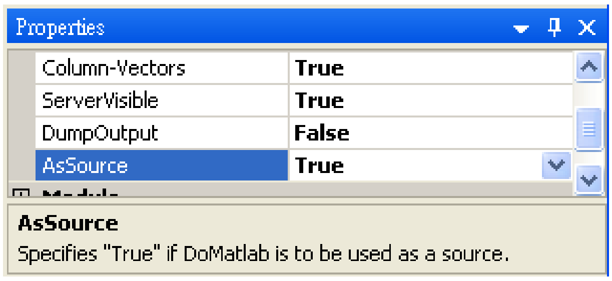
Option	Meaning
Complex	To output the analytic signal Z and data are saved in complex number.
SplitComplex	To split the real part X and the imaginary part Y of the analytic signal into two channels and save the values.
UnwrappedPhase	To obtain the phase angle, $\theta(t)$, corresponding to every time point. Unwrapped means that when the phase angle is bigger than 360° , it would not be wrapped into the range of 0° ~360°
InstantAmplitude	The amplitude values of analytic signal Z, $a(t)$
InstantFrequency	The differentiation of phase angle, $\theta(t)$, with respect to the time, of the analytic signal Z, i.e., the instantaneous frequency, $\omega(t)$

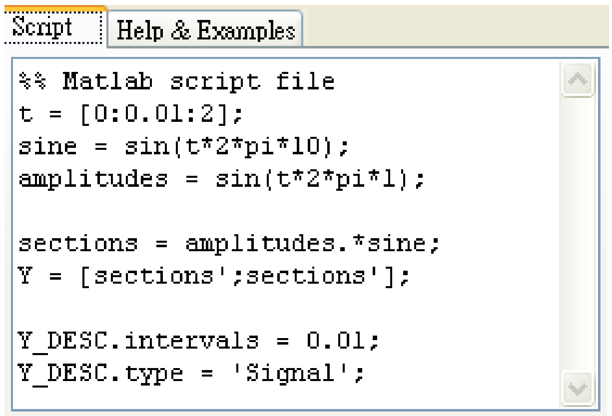
Example

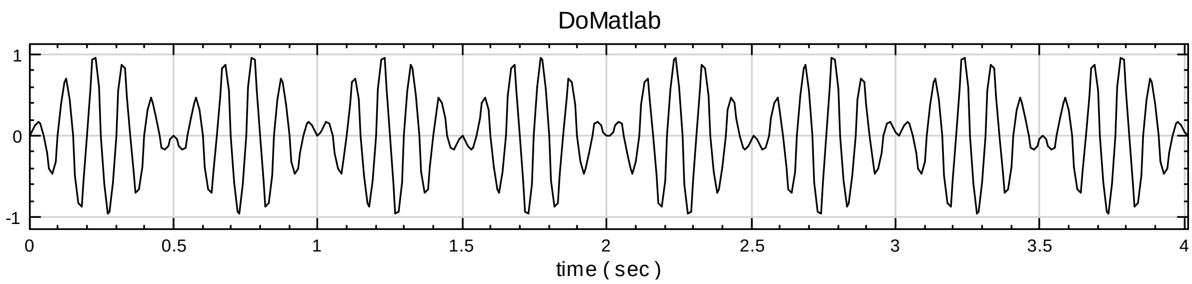
In this example, use Math→DoMatlab module to create a sine wave signal with amplitude of sine, perform Hilbert Transform, and show the meanings of property options by changing the *Output Type* of Hilbert Transform.

1. Select Math→DoMatlab to create a signal whose amplitude varies with time, i.e., $\sin(2\pi t)$. Perform dot product of amplitude and $\sin(2\pi t)$ to generate a sine wave whose amplitude is time-variant. The code of DoMatlab is shown below. Use *Viewer→Channel Viewer* to plot the result.

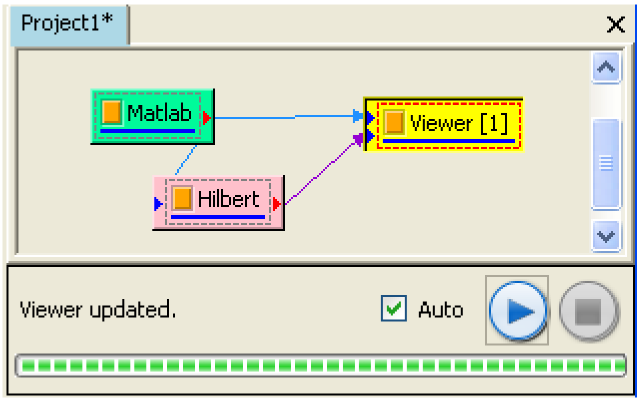


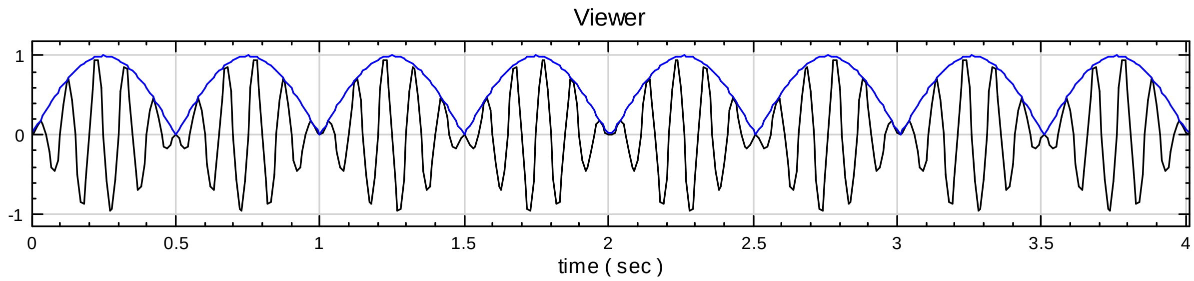




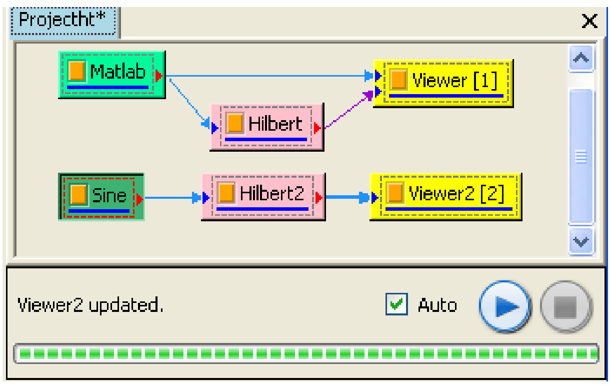


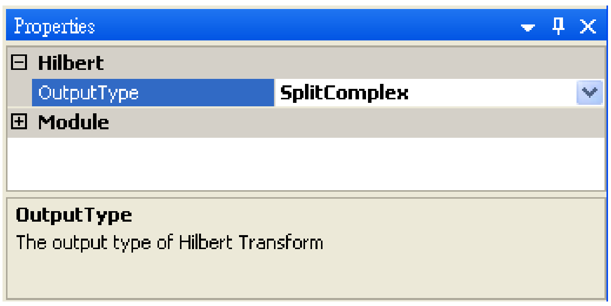
2. Select *Compute→Transform→Hilbert Transform* from DoMatlab to perform the calculation directly. The *Output Type* is the default value of Complex. Use *Viewer →Channel Viewer* to plot the result. Note that the output signal is complex while the default of *Properties/YValueType* in *Channel Viewer* is Magnitude, and therefore, the output shown in the Viewer is Magnitude of the output signal. After the calculation, connect the result to the same *Channel Viewer.* It can be seen that the result of Hilbert Transform is the upper envelop of the DoMatlab signal.

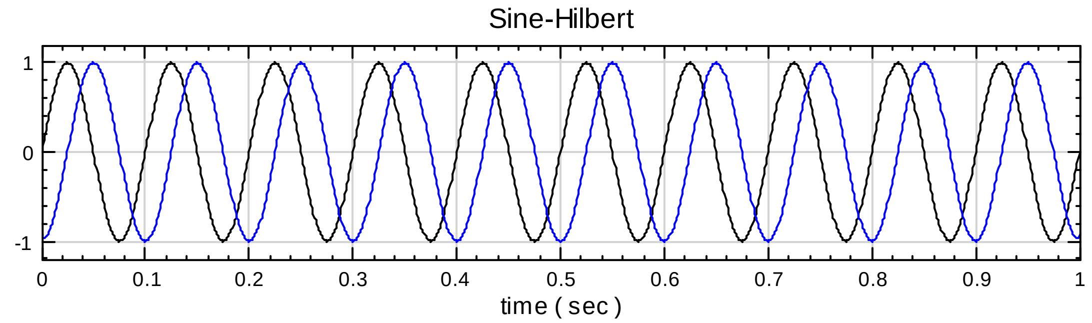




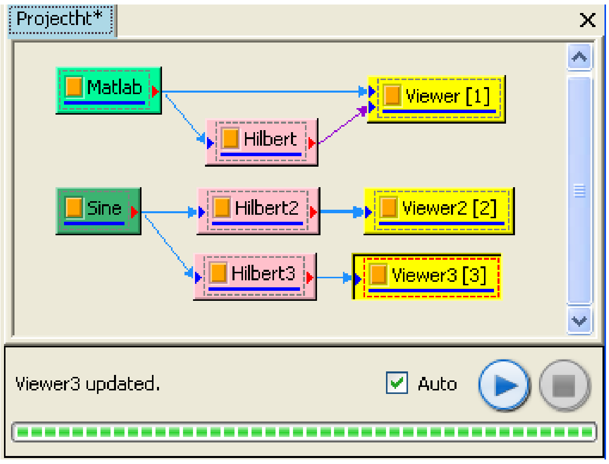
3. Next, create a sine wave source and perform Hilbert Transform. Change the *Hilbert→Output Type* in *Properties* to *Split Complex*. Hilbert Transform splits the real part and imaginary part into two channels. Use *Viewer→Channel Viewer* to plot the values of the real part and imaginary part, where the black curve is the real part and the blue curve is the imaginary part. It is shown that the imaginary part is 90° phase shift of the real part with respect to the phase angle.

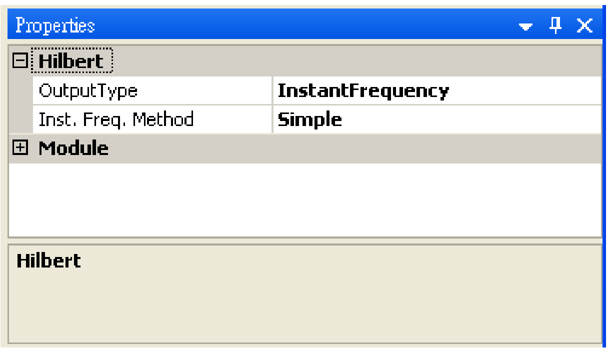


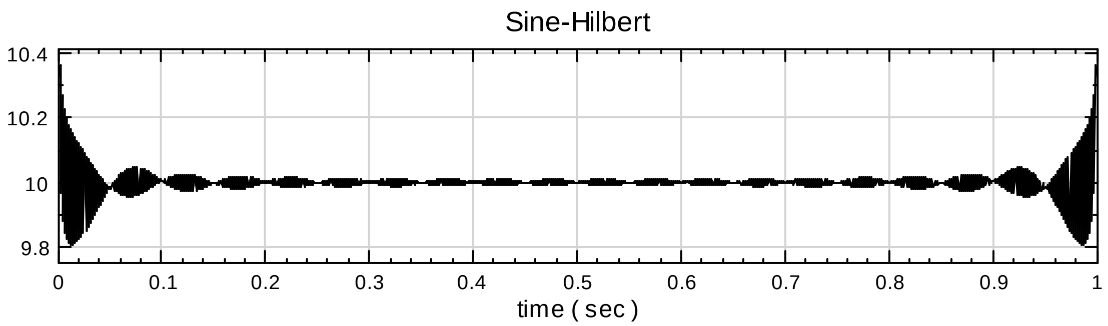




4. Repeat step 2 and change the *Output Type* to InstantFrequency only. The output would be the instantaneous frequency at every time point of the input signal.

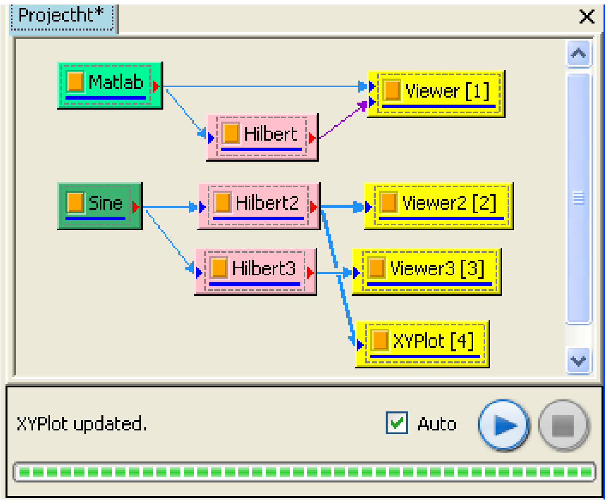


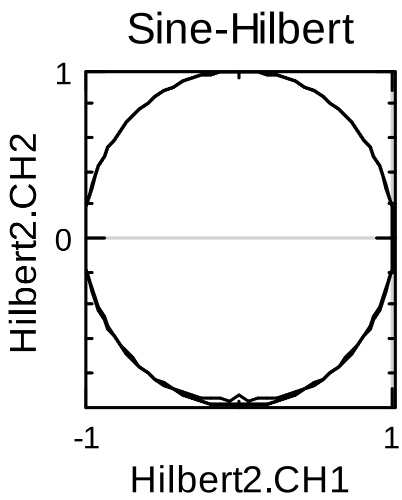




5. Channel Viewer is not able to plot the diagram in complex plane for analytic function Z(t). Viewer→XYPlot has to be used. Use Viewer→XYplot to plot the

output of *Split Complex* to obtain the diagram of analytic function Z(t) in complex plane. Changing the ViewerWidth and ViewerHeight to an equal value, i.e., 350, would make the proportion of x-axis and y-axis to be identical.





Related Functions

XYPlot, Hilbert Spectrum.

Reference

http://en.wikipedia.org/wiki/Hilbert_transform

3.6.5 AutoCorrelation

AutoCorrelation is defined as the convolution of a signal with itself and it can be used for correlation analysis. AutoCorrelation analysis can reveal signal periodicity and how fast a signal varies in time.

Introduction

Mathematically, the definition of autocorrelation is

$$R^{XX}(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_0^T x^*(t)\cdot x(t+\tau)dt$$

where $x^*(t)$ is the conjugate of $x(t)$, T is the signal period and τ is the time delay.

In the discrete case, let $X(t)$ be a N-length time series, where t is time coordinate, the autocorrelation formula is

$$R^{XX}(\tau) = \frac{1}{N-\tau}\sum_{t=0}^{N-1} x(t)x(t+\tau), \ -(N-1)\le\tau\le(N-1)$$

$R^{XX}(0)$, i.e., $(\frac{1}{N}\sum_{t=0}^{N-1} X^2(t))$, is equivalent to mean square value Ψ_X^2

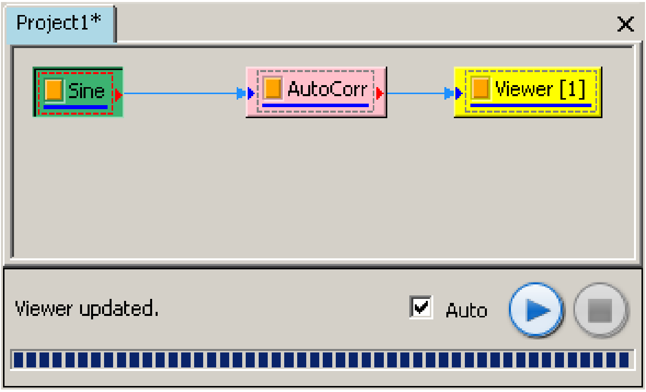
Properties

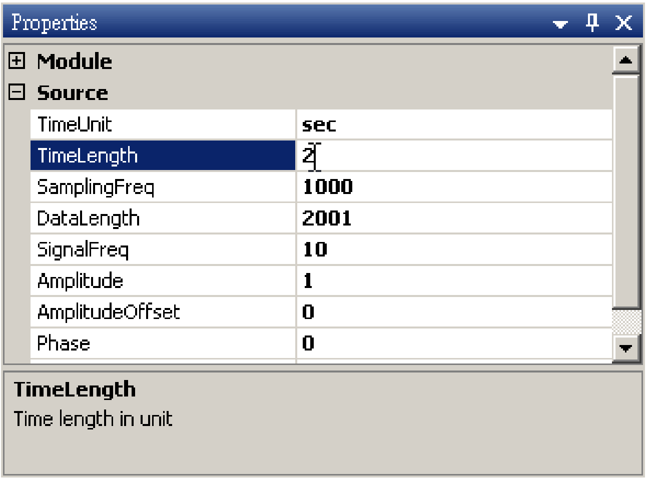
This module accepts input of Signal (which could be real number, single channel, Regular) and Audio (which could be real number, single channel, Regular). The output format is real, signal channel Regular signal. This module does not have any other properties to be set.

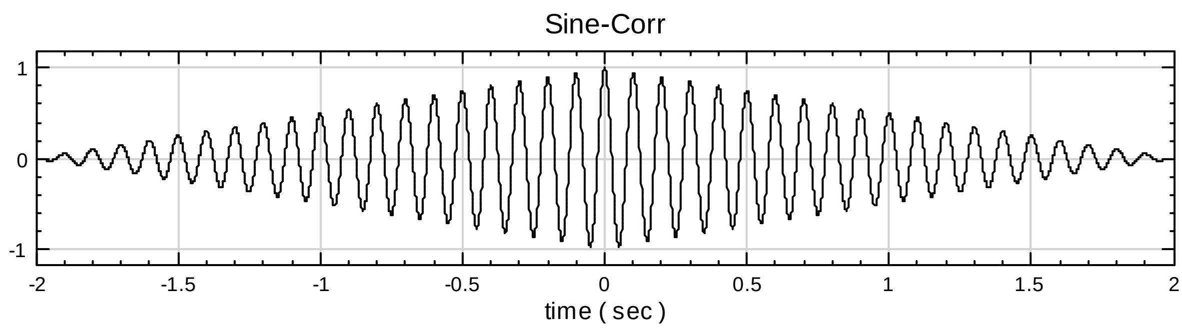
Example

In this example, perform auto correlation on a sine wave, a white noise and the mixed signal of these two signals.

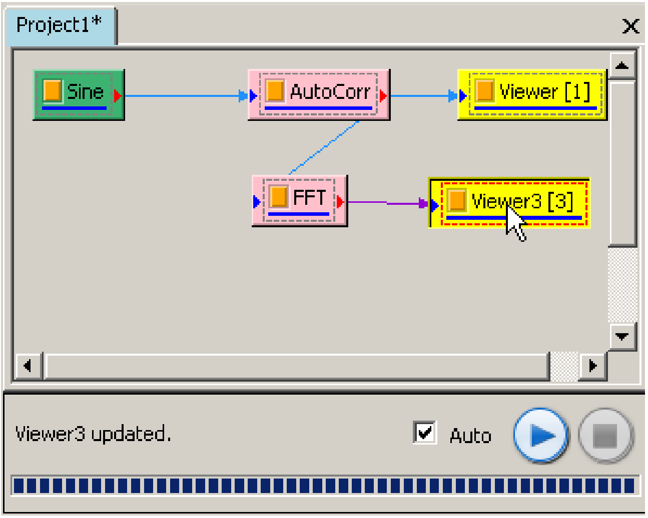
1. In Network Window, select *Source→Sine* to create a sine wave and change its *Properties/*DataLength to 2 seconds and keep the frequency to 10 Hz. Perform *Compute→transform→AutoCorrelation* on this signal and then plot the result using *Viewer→Channel Viewer*.

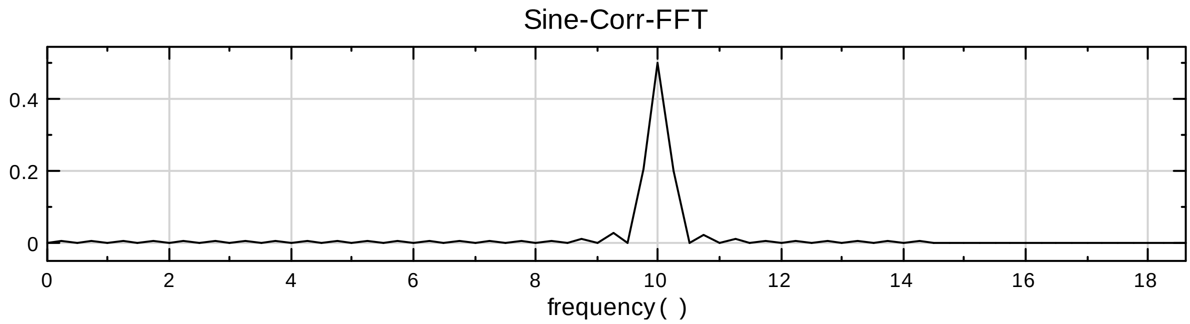




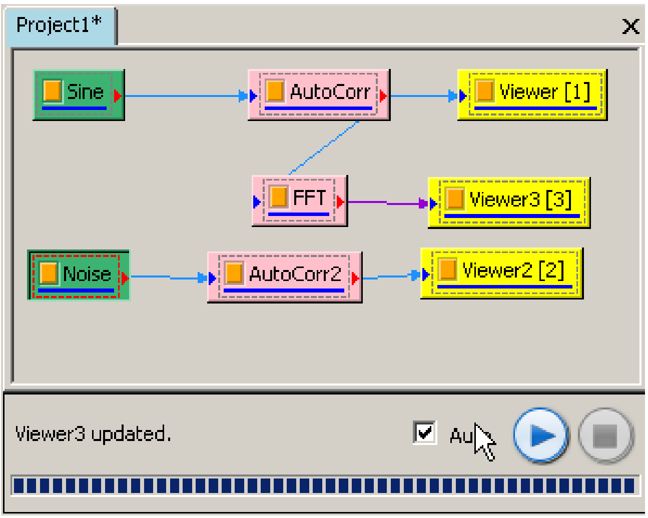


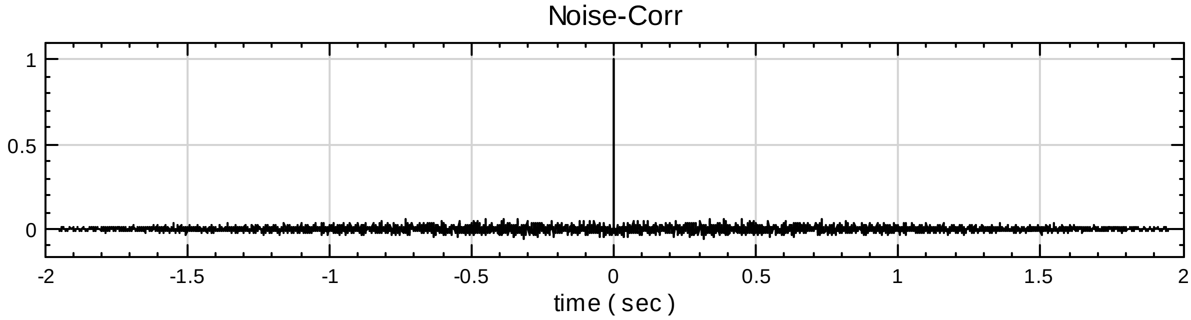
2. The figure above shows the auto correlation of the sine wave with different delay, i.e., the τ in the original formula. Perform FFT transform on the output of AutoCorr. It is shown that the frequency of the output signal of AutoCorr is 10. This verifies that the auto correlation of a periodic signal preserves the frequency of the input signal.



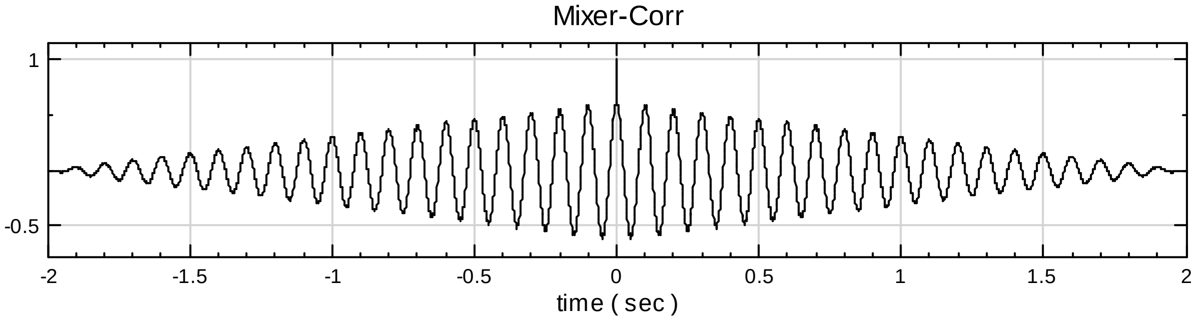


3. Create a White noise and then repeat step 1. The result is shown below.





4. Use *Compute→Mathematics→Mixer* to mix the sine and white noise, and then perform the auto correlation operation. The result is shown below.



Related functions

Cross Correlation, Source, Mixer.

Reference

http://en.wikipedia.org/wiki/Autocorrelation

3.6.6 CrossCorrelation

Cross Correlation performs convolution on two time series for cross correlation analysis. In order to reveal the characteristics of an unknown signal, the common practice is to perform cross correlation on this unknown signal with a well-known signal.

Introduction

Mathematically, CrossCorrelation is defined as

$$(x * y)(t) \equiv \int_{-\infty}^{\infty} x^*(t)y(t+\tau)d\tau$$

where τ is the time delay.

In discrete case, let $X = \{x_0, x_1, x_2 \cdots x_{N-1}\}$, $Y = \{y_0, y_1, y_2 \cdots y_{M-1}\}$ to be a N-length and a M-length time series, respectively, the cross correlation with time delay j is given as

$$R_j^{XY} = \sum_{i=0}^{N-1} x_i \; y_{i+j}$$

The length of the result of cross correlations is N+M-1. R^{XY} is asymmetric and its maximum value is subject to the inequality below

$$R^{XY}(0) \leq \sqrt{R^{XX}(0) \cdot R^{YY}(0)}$$

If the two input signals are independent events statistically, then $R_k^{XY} = R_k^{YX}$. The procedure of calculating the cross correlation of a signal itself, i.e. R_k^{XX}, is the Auto Correlation.

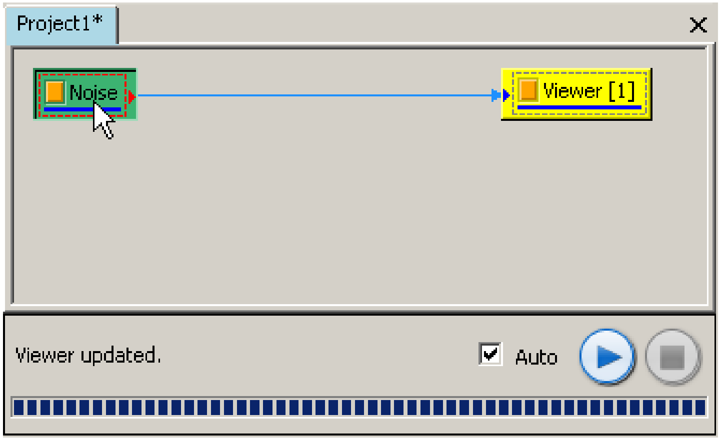
Properties

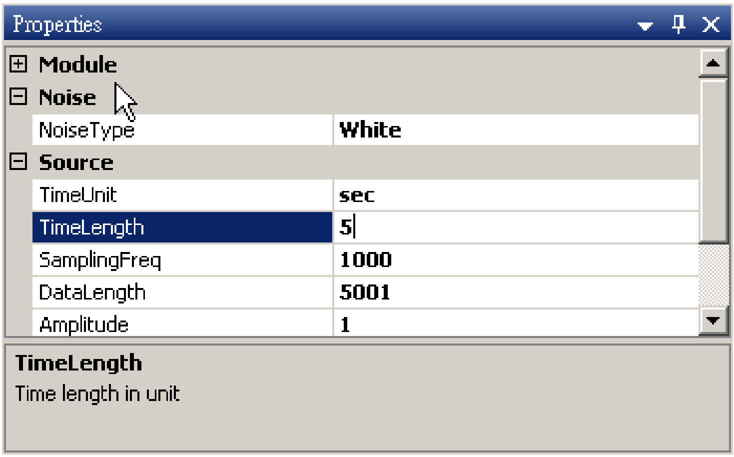
This module accepts input of Signal (which could be real number, single channel, Regular) and Audio (which could be real number, single channel, Regular). Two input signals are required. The output format is real, single channel, Regular signal. The lengths of the two input signals could be different. However, the sampling frequency and time unit of these two signals must be identical.

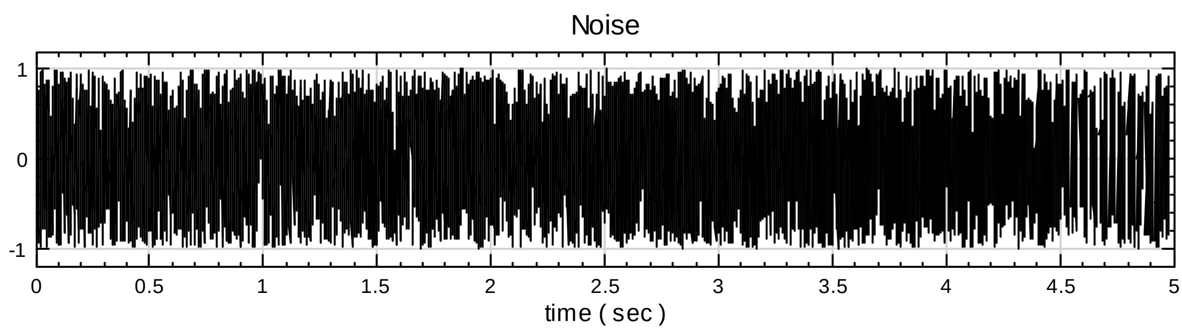
Example

First, create a white noise as original signals and then use part of the data to create another noise signal. Next, perform Cross Correlation on these two signals and observe the relationship between them.

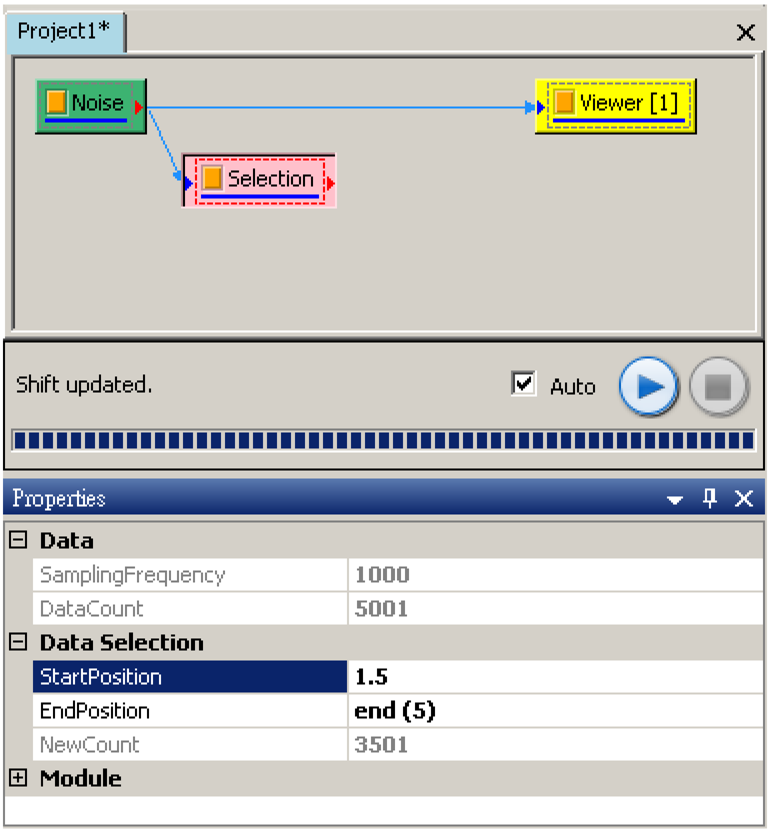
1. Select Source→Noise to create a noise. Change the *Properties/TimeLength* to 5 sec and use *Viewer→Channel Viewer* to plot the result.



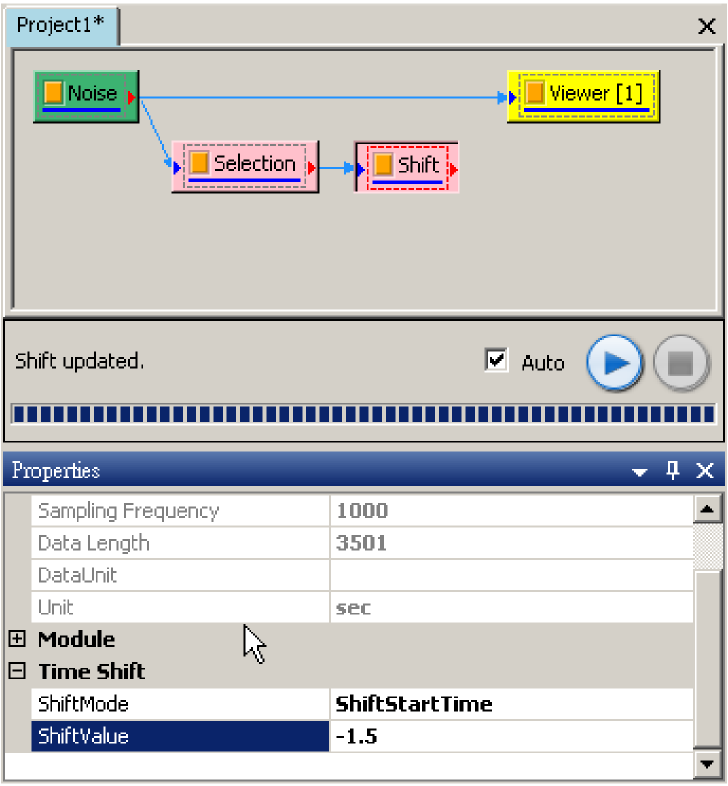


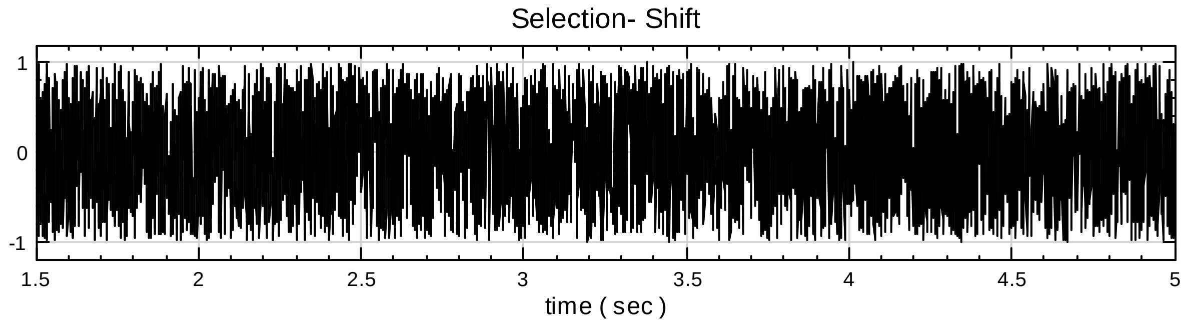


2. Click on the icon of the Noise and then press right mouse button to select *Compute→Channel→Data Selection*. Change *Properties/EndPosition* field to 3.5.

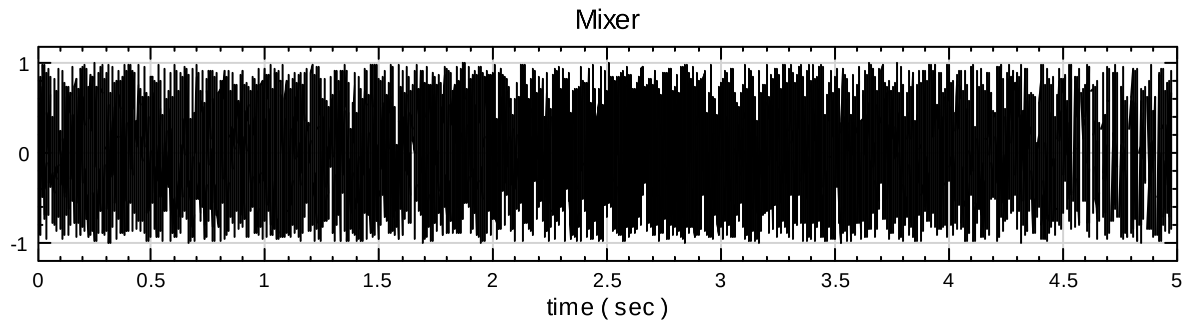


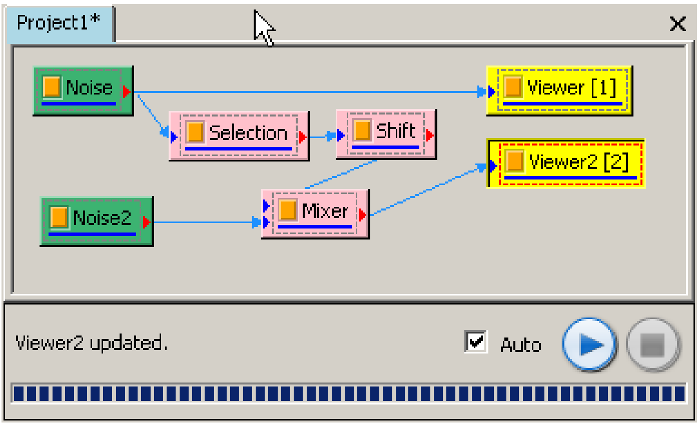
3. Select the *Compute→Channel→Time Shift* to connect with the *Selection* SFO. Change *Properties/Shift Value* to -1.5 which shifts the data in 0~3.5 seconds to 1.5~5 second.



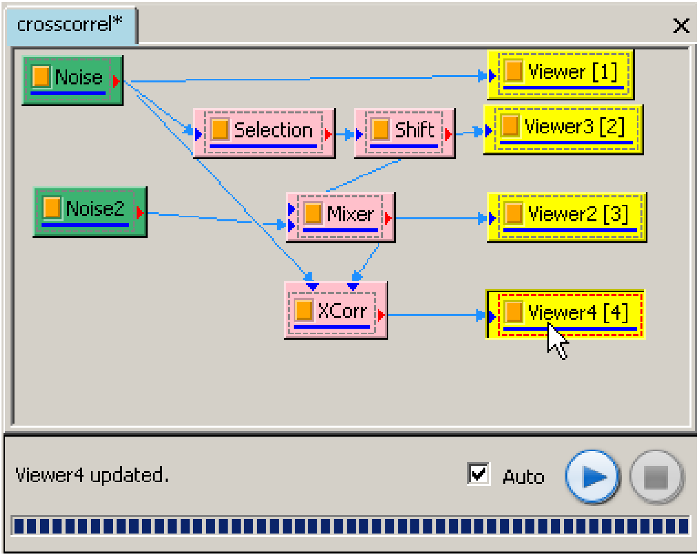


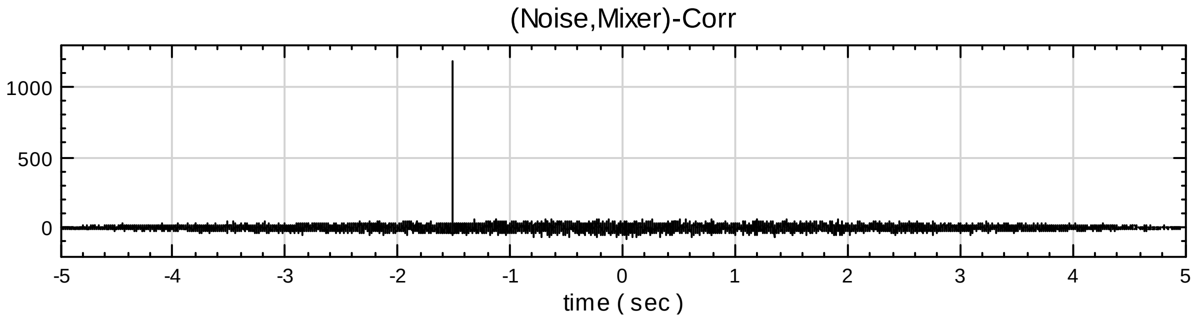
4. Create another noise signal that the TimeLength is 1.5 seconds. Use *Compute→Channel→Mixer* to mix the original noise and the one processed by *Time Shift* to obtain the second signal. And then use *Channel Viewer* to plot the result.





5. Perform Cross Correlation on the Noise signal and the output signal from the Mixer. Then use *Channel Viewer* to plot the result.





The above figure shows that the strong correlationship between siganl 1 and signal 2 at -1.5 sec. Essentially, the data of 1.5~5 second in the 2nd signal and the data of 0~3.5 second in the 1st input signal are identical. Therefore, this result is consistent with the characteristics of these two input signals.

Related Functions

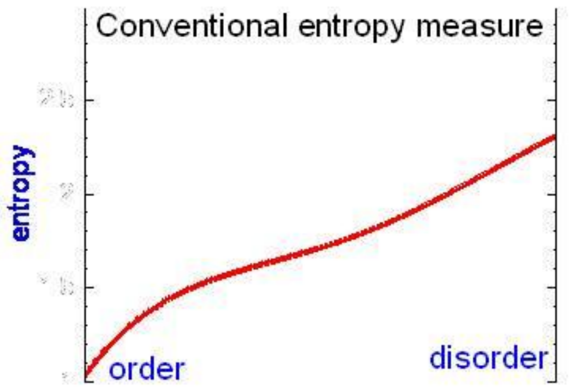
Auto Correlation, Data Selection, Mixer, Time Shift.

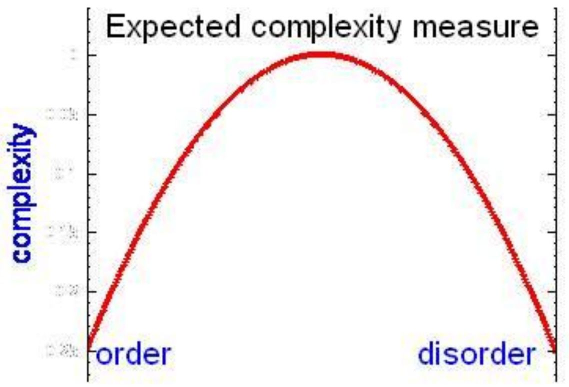
3.6.7 MSE

MSE is the abbreviation for Multiscale Entropy. It analyzes the complexity of time series of the system. The complexity value represents the adaptability of the system to the environment. The higher is the complexity, the healthier is the system. This method has shown good results in biological systems, earth science, and mechanical vibration.

Introduction

MSE is different from traditional Entropy method. In traditional Entropy theory, the more chaotic is the system, the higher is the entropy value. For the theory of MSE method, the complexity represents the heath of the system. If the system is too uniform or too chaotic, the system deems to be unhealthy. MSE is the entropy value at every scale level. The complexity represents the entropy changes under different scales.





This module uses Sample Entropy to calculate the entropy value. It is an improved version of the Approximate Entropy. It requires less data points than Approximate Entropy does.

For the concept of the scale, we can define the scale τ of the time series $y_j^{(x)}$ as

$$f_j^{(x)} = \frac{1}{\tau} \sum_{i=(j-1)x+1}^{jx} f_i \, , 1 \le j \le \frac{N}{\tau}$$

Every point $f_j^{(x)}$ in the series is the average value within each scale. We can calculate Sample Entropy for every scale τ:

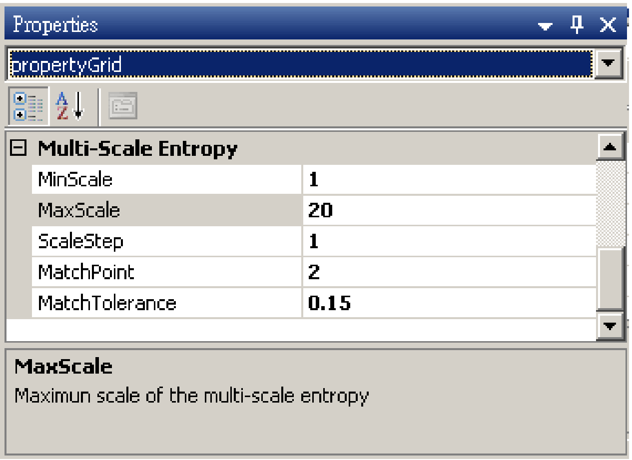
$$S_E(scale\,,\, m\,,\, r\,,\, N) = \ln\left(\frac{\sum_{i=1}^{N-m} n_i^m}{\sum_{i=1}^{N-m-1} n_i^{m+1}}\right) = \ln\left(\frac{n_n}{n_d}\right)$$

where n_i^m satisfies $d[f_m(i)\,,\, f_m(j)] \le r$, and d is the distance of the Euclidean geometry:

$$d[f_m(i)\,,\, f_m(j)] = max\{\left|f(i+k) - f(j+k)\right|\,,\, 0 \le k \le m-1\}$$

Properties

This module accepts input signals of real number, single channel or multi-channel, regular, and audio. The format of the output is real number, single channel or multi-channel, and regular. The properties are defined below.

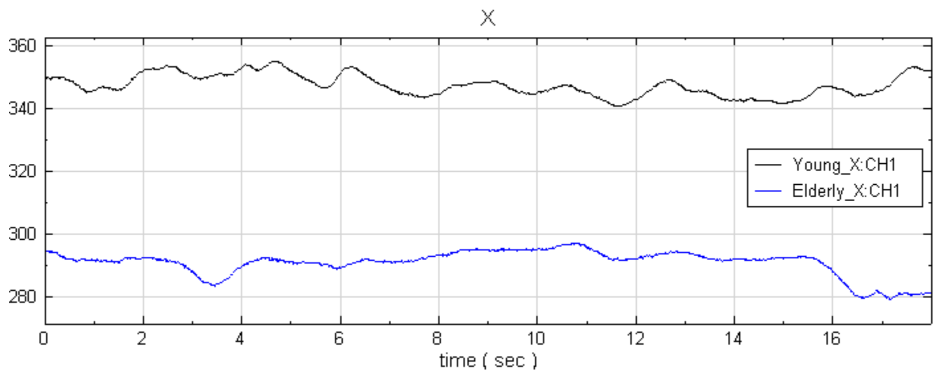


Property Name	Property Definition	Default Value
MinScale	The minimum scale	1
MaxScale	The maximum scale	20
ScaleStep	The increment step of scale during calculation, from the minimum to the maximum increased (decreased) by this step.	1
MatchPoint	Set the length of the pattern for comparing	2
MatchTolerance	Set the tolerance of pattern comparison. This is r in the original equation.	0.15

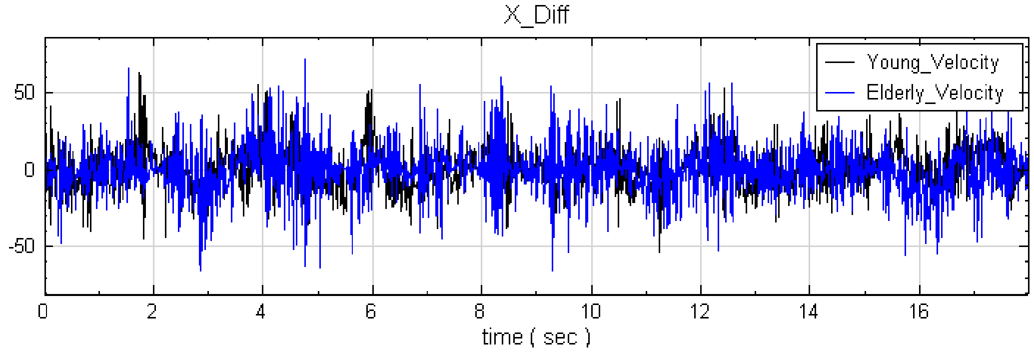
Example

This example is for physiological signal.

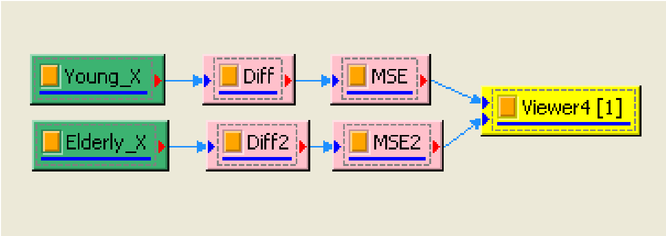
In the Project panel, load two data sets. They are the recordings of the standing Center of Pressure (COP) along X axis from young and elderly subjects respectively. Then use Viewer / Channel Viewer to display the data.

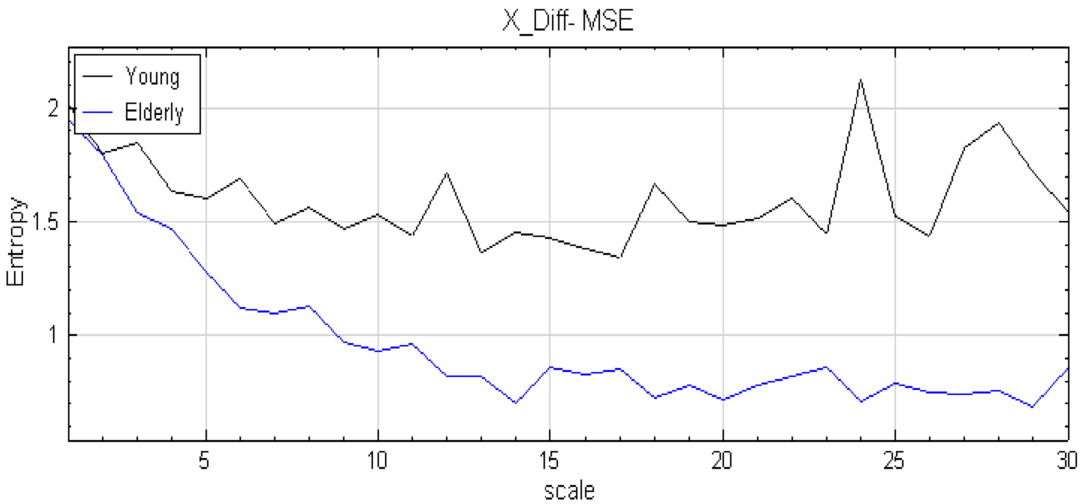


To understand the ability of the subjects to keep their balance, we first take the differential values of the original data. It is the velocity of COP which is shown below.

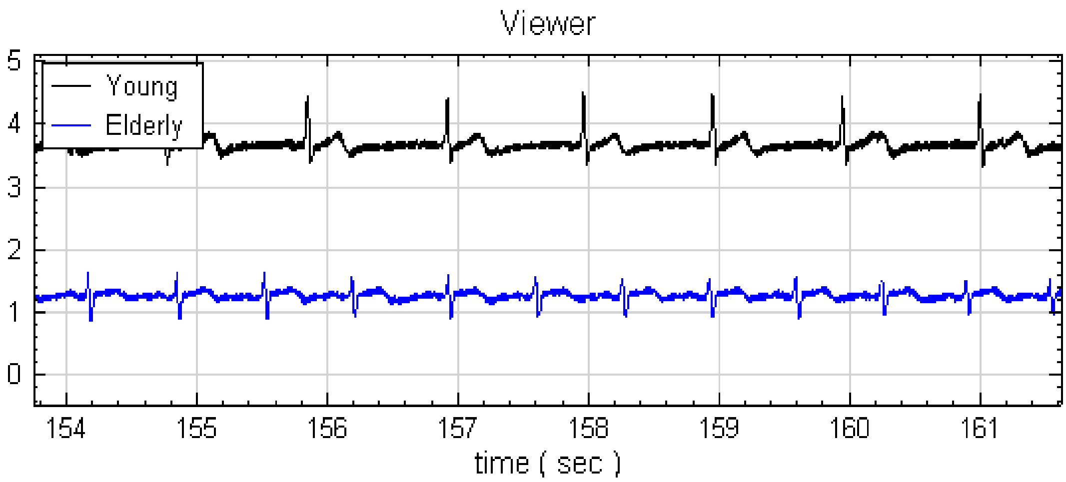


Then input velocity signals to Compute / Transform / MSE for calculation, and display the results with Viewer / Channel Viewer. It shows that the balance of the elderly is not as good as the young subject. The change of elderly's COP is not adaptive; the complexity value becomes lower when the scale increases. However, for young adults, their balance can adapt to changes quite well. The complexity value stays the same when scale increases.

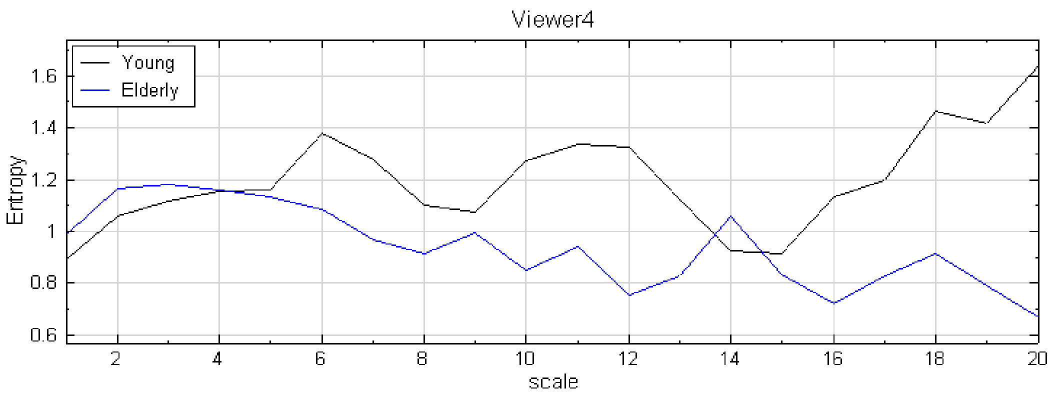




Here is another example. The ECG recordings of 18 and 55 year-old healthy subjects are shown below.



Calculate heart beat intervals of both recordings, i.e. the RRinterval, then apply MSE method, the results are shown in the graph. It is shown that the complexity of the young is higher than the one of the elderly. The heart of the young subject has better ability to handle external stimulus and adjust to the pressure accordingly.



Related Functions

Noise, Viewer.

References

Pincus, S. M., Approximate entropy as a measure of system complexity, Proceedings of the NationalAcademy ofSciences, USA, Vol. 88, pp. 2297-2301 (1991).

Costa M., Goldberger A.L., Peng C.-K. Multiscale entropy analysis of physiologic time series. Phys Rev Lett 2002; 89:062102.

Costa M, Peng C-K, Goldberger AL, Hausdorff JM. Multiscale entropy analysis of human gait dynaiics. Physica A , 2003;330:53-60.

Costa M., Goldberger A.L., Peng C.-K. Multiscale entropy analysis of biological signals. Phys Rev E 2005;71:021906.

3.7 HHT (Hilbert-Huang Transform)

Hilbert-Huang Transform (HHT) is an empirical signal processing method which can be used to reveal true physical meanings for non-stationary and non-linear signals.

Most traditional data processing methods are based on linear and stationary assumptions. Only in recent years have new mathematical methods been developed to process either the non-stationary data or the nonlinear data. However, in a lot of systems in the real world, the data are most likely to be both nonlinear and non-stationary. Analyzing the data from such a system is a daunting problem. To resolve such a problem, Dr. Norden Huang at NASA has developed a new mathematic method: Hilbert-Huang Transform (HHT).

The HHT method consists of two parts: empirical mode decomposition (EMD) and Hilbert spectrum. By combining EMD and Hilbert Spectrum (in the component of Compute→TFA) or Hilbert Transform（in the component of Compute→Transform）, this method is viable for nonlinear and non-stationary data analysis, especially for time-frequency-energy representations. In most researches, HHT can provide more information about the relationship of time, frequency, and energy. And in most cases, HHT can reveal true physical meanings of the system, explain physical phenomenon and solve engineering problems.

Introduction

EMD (Empirical Mode Decomposition) is a mathematical method which can be used to decompose a signal into several Intrinsic Mode Functions (IMF) and a residue. IMF can be viewed as a generalized Fourier Transform. The time-varing amplitude $a_j(t)$ and the instantaneous frequency $\omega_j(t)$ have not only greatly improved computation efficiency, but also make it possible to extract non-linear and non-stationary characteristics from signals. With IMF, the amplitude and the frequency modulations are also clearly separated. Thus, the restriction of the constant amplitude and fixed frequency of the Fourier transform has been overcome, with a variable amplitude and frequency representation.

This component decomposes a raw signal into several sub-signals based on signal characteristics by EEMD. In contrast to other mathematical methods, EEMD is empirical and intuitive. This method is based on an assumption that any data consist of different simple intrinsic modes of oscillations, called Intrinsic Mode Function (IMF). Every IMF includes characteristics listed below.

1. The number of extrema and the number of zero-crossings either be equal or differ at most by one
2. At any point, the mean value of envelope defined by the local maxima and the envelope defined by the local minima is zero (i.e., symmetric)
3. No constant amplitude and fixed frequency

In EEMD, **Sifting process** is used to extract IMF from signals. The procedure is as follows.

1. Take a signal $x(t)$, identify all the local extrema, then connect all the local maxima by a cubic spline to obtain the upper envelope
2. Repeat the procedure in step 1 for the local minima to produce the lower envelope
3. Calculate the mean of the upper and lower envelope to obtain a mean curve, which is designated as $m(t)$
4. Subtract the original signal with the $m(t)$ and let the result to be $h(t)$, i.e., $h(t)=x(t) – m(t)$

The **Sifting process** stated above is defined as a process to obtain $h(t)$ using the $m(t)$ to subtract the mean obtained from upper and lower envelope.

With the Sifting procedure, EEMD processing method is given as follows:

After being processed by sifting several times, the mean of upper and lower envelope of an original signal should gradually move close to and overlap with the x-axis. The first IMF will be obtained when the upper and lower envelop is symmetric by the x-axis. The first IMF is designated as IMF1. Subsequently, subtract the IMF1 from the original signal to obtain the First Residue, designated as *r1*. Repeat the same procedure for the *r1* to obtain IMF2. Then, subtract *r1* from IMF2 to get the second Residue, *r2*. Conduct the same procedure repeatedly; the original signal can be decomposed of several IMF and a final Residue.

There are two types of Sift Iteration and Stoppage Criterion for EEMD,

1. Cauchy type of convergence test. Specifically, when the normalized squared difference (SD_k) between two successive sifting operations (defined as below) is less than a predetermined value.

$$SD_k = \frac{\sum_{t=0}^{T} \left| h_{k-1}(t) - h_k(t) \right|^2}{\sum_{t=0}^{T} h_{k-1}^2(t)}$$

2. A number *S* is pre-selected. The sifting process stops after *S* consecutive times or when the numbers of zero-crossings and extrema are equal to or differ at most one.

The two criterions above are used simultaneously and the sifting process will be terminated when either is satisfied. DataDemon outputs the decomposing results in the order of high frequency to low frequency, i.e., the channel 1 is the highest frequency, the channe2 is the next, and so on, until the last channel which outputs the Residue.

There are several functions associated with HHT module:

1. RCADA EEMD: EEMD method from Research Center for Adaptive Data Analysis (RCADA) at National Central University in Taiwan
2. RCADA Instant Frequency: A method to calculate instant frequency from RCADA
3. RCADA Spectrum: Hilbert Spectrum method from RCADA
4. IMFProperty: List properties for each IMF, including the number of zero crossing, the number of maxima, average frequency of zero crossing, orthogonality among IMFs, power ratio of each IMF etc.

3.7.1 RCADA EEMD

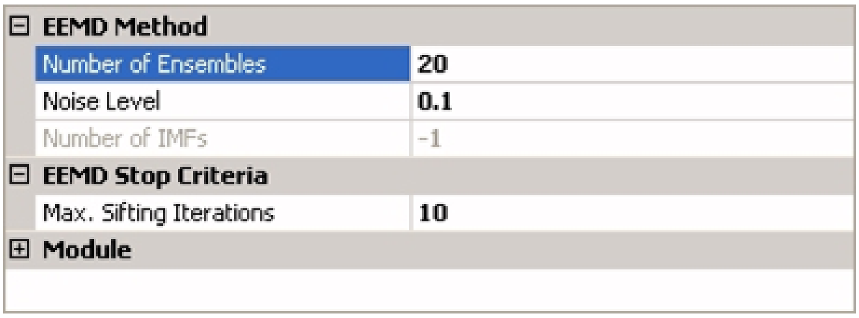
RCADA EEMD is the latest (2009) algorithm from Dr. Norden Huang which is published at Research Center for Adaptive Data Analysis (RCADA) at National Central University in Taiwan. This module has the same algorithm as the MATLAB code published at RCADA; however, it provides more parameters, such as boundary condition and random number generator. The result obtained from the module is the same as the one from RCADA MATLAB code. And the module runs at least 200 times faster than RCADA MATLAB code.

Introduction

Please refer to http://rcada.ncu.edu.tw/research1.htm for details.

Properties

This module accepts input signal of real number, single channel, Regular and Audio. The properties are introduced below.



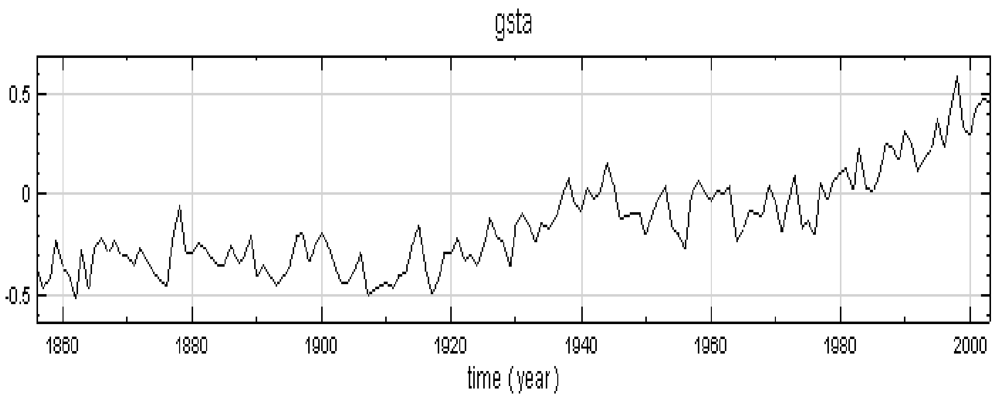
Property Name	Property Definition	Default Value
SplineType	Set the boundary condition: Clamped Spline, Nature Cubic Spline, or Not A Knot. Detailed in the table below.	NotAKnot
Number of Ensembles	The realization number of adding noise for EEMD. The resulting IMF of EEMD is the average of corresponding IMF from all realizations. For example, if Ensumble number is 20, there are 20 runs of EMD for the original signal with added noise. The final IMF is the average of these 20 groups of IMFs.	20
Noise Level	The amplitude of noise adding to the signal. It is the percentage of the standard deviation of the original signal.	0.1
Max. Sifting Iterations	The maximum number of sifting procedure	10

Number of IMFs	The max number of IMF channel	-1
Modify Envelope Endpoints	Set the endpoint of envelop to use outer differetial calculation	True
Use Guassian Noise	Use Guassian Noise; otherwise use White Noise	Fasle

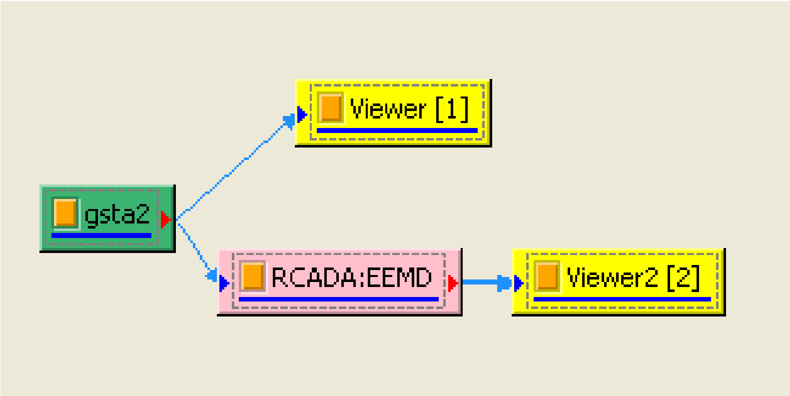
Boundary Condition	**Defintion**
NotAKnot	The 3rd derivative of points on the boundary (the closest) is equal. This is the same condition as in the MATLAB code at RCADA. However, the final results may differ due to different random noise added.
NatureCubicSpline	The 2nd derivative of the points on the boundary is 0.
ClampedSpline	The 1st derivative of the points on the boundary is fixed (=0).

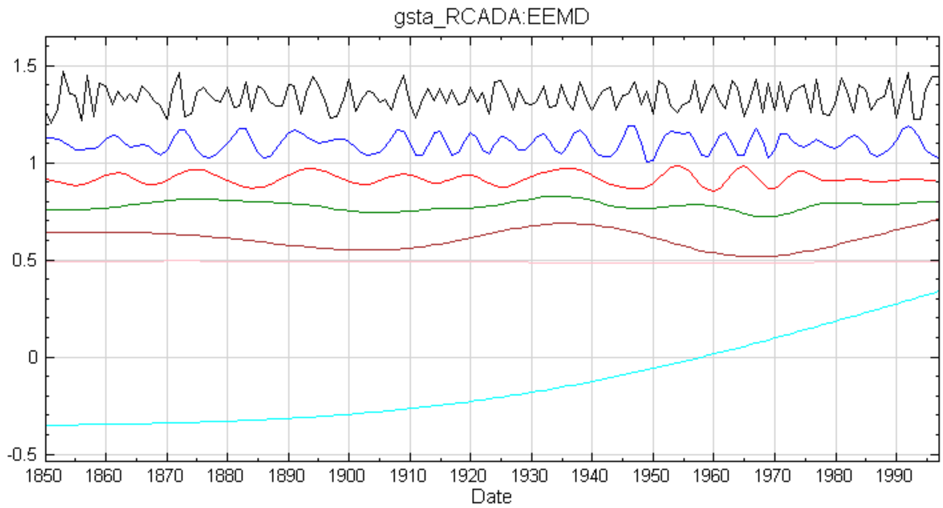
Example

This example uses the data gsta.dat from RCADA. It is the annual average temperature on earth surface. You can refer to demo68 in C: \ Program Files \ DynaDx \ DataDemon \ demo \ HHT.

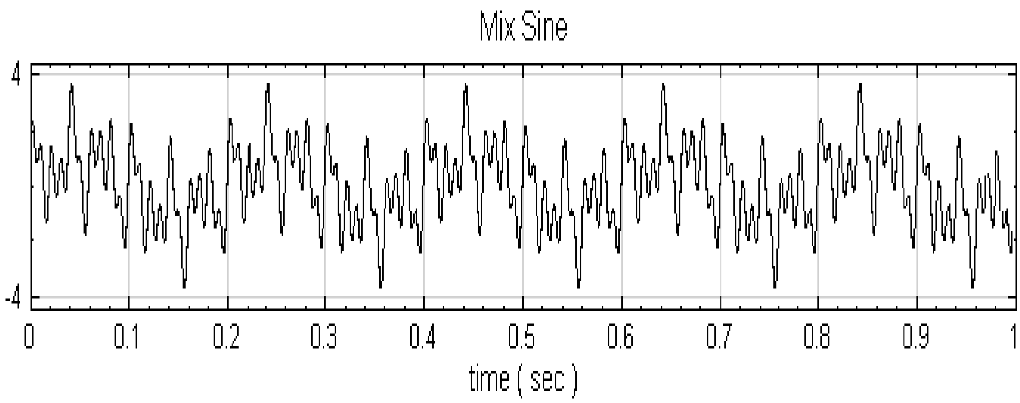


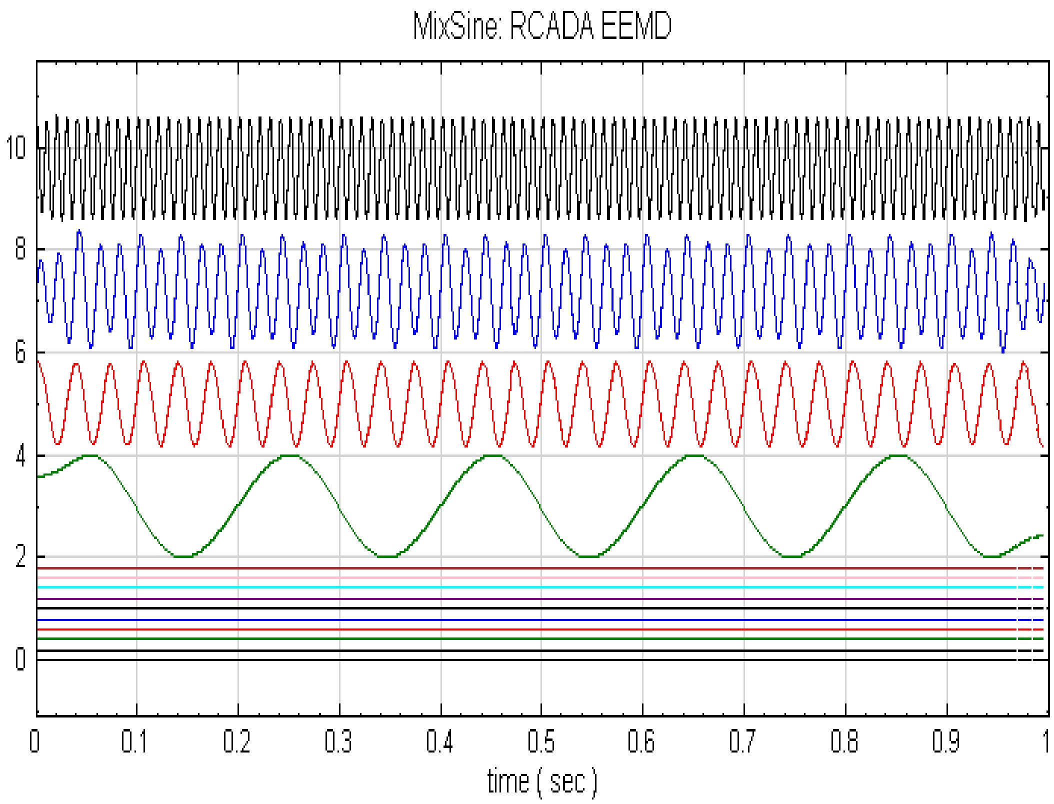
Connect the signal to Compute / HHT / RCADA EEMD for decomposition. With default parameter setting, the results are shown below. Note the last curve in light blue is the Residual and it shows the temperature trend.





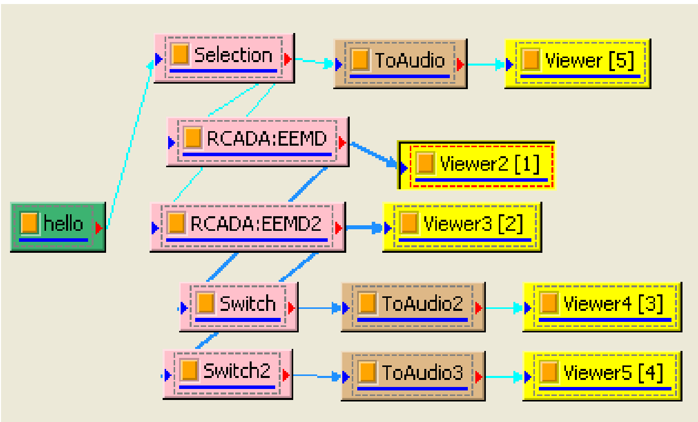
Let us compare the running speed with MATLAB code from RCADA using a same signal. The signal is $sin(5 \cdot 2\pi t) + sin(30 \cdot 2\pi t) + sin(50 \cdot 2\pi t) + sin(100 \cdot 2\pi t)$ and the length of this signal is 20,000. The result of RCADA EEMD calucation is:

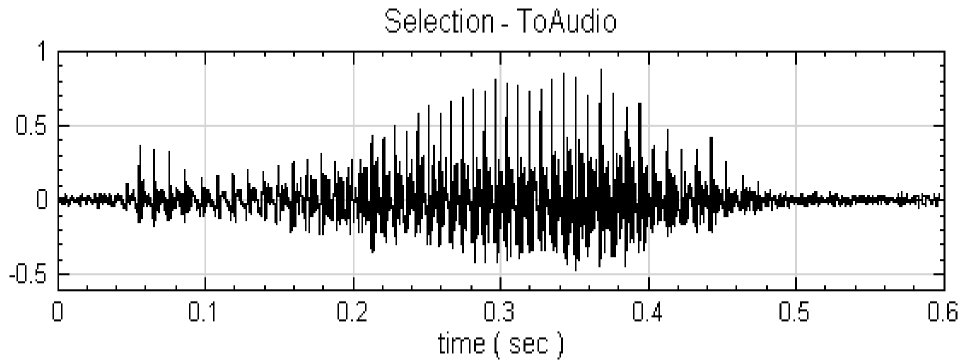




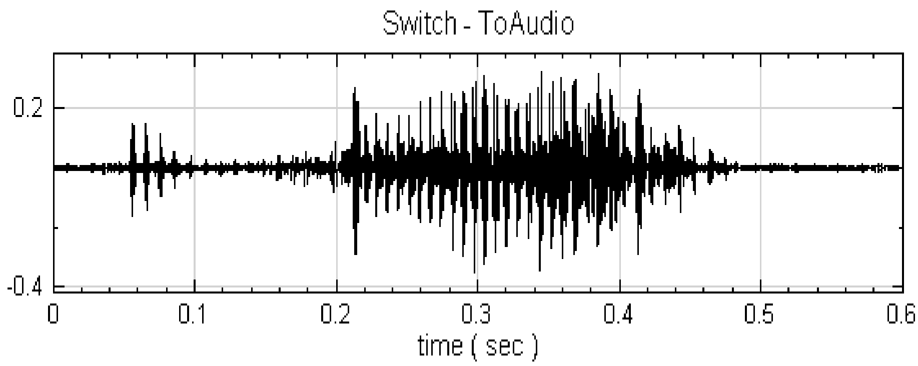
On an Intel Dual Core E6300 (2.8GHz) computer, it takes 1.65 seconds to complete the calculation for DataDemon. However, it takes 376.60 for MATLAB to finish. The difference is more than 200 times.

Refer to demo68_1 in C: \ Program Files \ DynaDx \ DataDemon \ demo \ HHT. The signal is “Hello” voice wave.

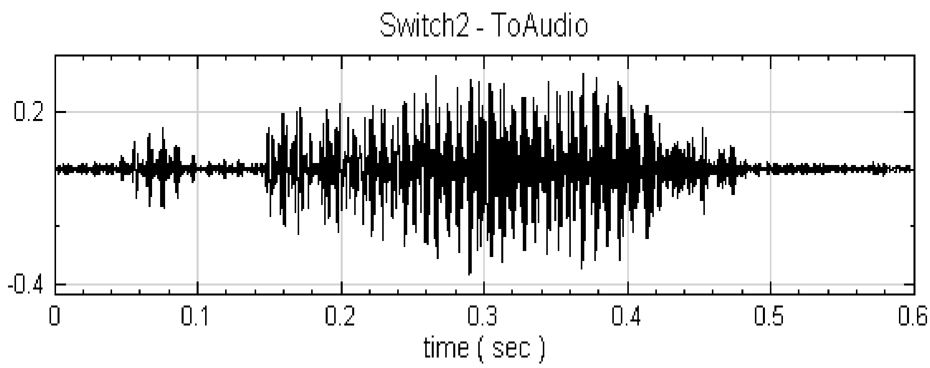




In RCADA : EEMD properties, set Number of Ensembles to 200 and Noise Level to 0.25. Observe the signal in the 3rd channel. The result is shown below:



In RCADA : EEMD2 properties, set Number of Ensembles to 1 and Noise Level to 0.1. Observe the signal in the 3rd channel. The result is shown below:



After playing the sound after EEMD decomposition, it shows that the result of the 1st EEMD sounds clearer and the result of the 2nd EEMD has noises.

Related Functions

RCADAInstant frequency, RCADA Spectrum.

References

Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung and Henry H. Liu: “The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-Stationary Time Series Analysis”, Proceedings of the Royal Society, Vol. 454, No. 1971, 1998.

Huang, N. E., M. L. Wu, S. R. Long, S. S. Shen, W. D. Qu, P. Gloersen, and K. L. Fan (2003): “A confidence limit for the Empirical Mode Decomposition and Hilbert Spectral Analysis”, Proc. Roy. Soc. London, 459A, 2317-2345.

ZhaoHua Wu and Norden E. Huang, 2009, “Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method”, Advances in Adaptive Data Analysis, Vol. 1, No. 1 (2009) 1-41.

http://rcada.ncu.edu.tw/research1_clip_reference.htm

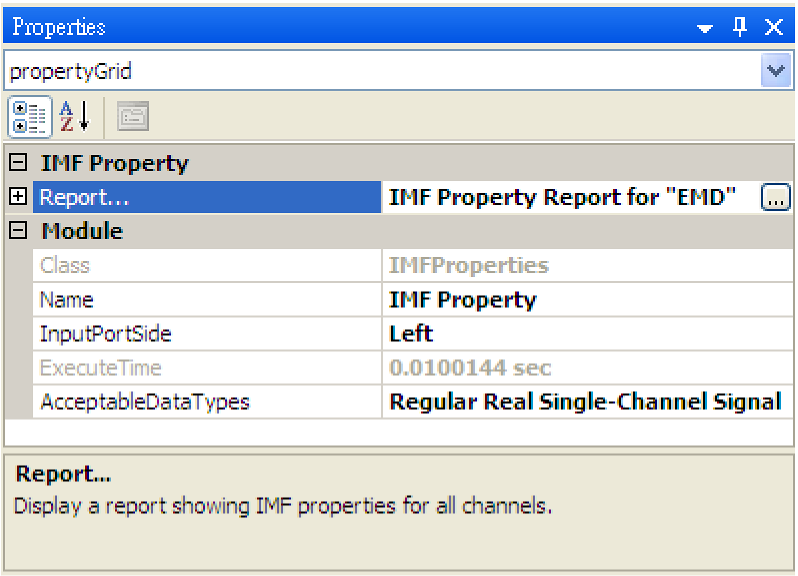
3.7.2 IMF Property

A signal can be decomposed into several IMFs and a residue after processed by EEMD. The functionality of IMF Property is to show characteristics of each IMF and validate the decomposition from EEMD.

Introduction

IMF decomposed by EEMD has one characteristic that the number of extrema and the number of zero-crossings either be equal or differ at most by 1. To validate the decomposition result of EEMD, we can use this module to show characteristics of each IMF. For every IMF, this module lists the number of zero-crossings and extrema, the average frequency of zero-crossing, the orthogonal relationship between IMFs and the power ratio.

Properties



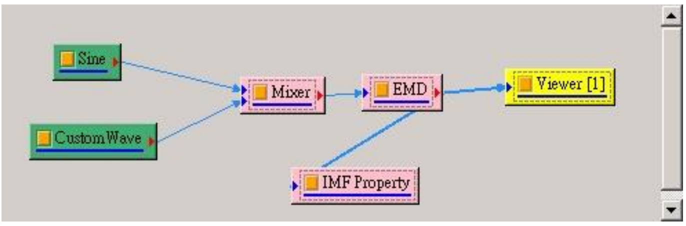
Property Name	Property Definition
General properties	For every IMF, show the number of Zero Crossings, the number of Extrema (Extreme Counts), and the average instantaneous frequency. The average instantaneous frequency is estimated by the number of zero crossings and the number of extreme.
Orthogonality	To calculate all IMF orthogonal matrix.
Percentage Power	To calculate the energy percentage of every IMF

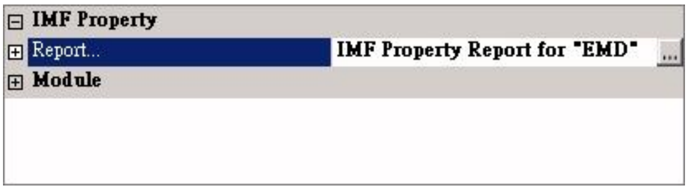
	compared to the total energy of the original signal, except for Residue.

Example

Following the example of HHT, use the IMF Property component to show the signal characteristics after decomposed by EEMD operation.

Right mouse button click on EEMD SFO and select *Computer→HHT→IMF Property* as shown below. For every IMF analyzed, after clicking the *Report…* button, which locates on the right, a result window is poped up as shown below.





General 3

IMF Property Report for "EMD"

General Properties

Channel	Zero Crossings	Extreme Counts	Average Frequencies
IMF_h1	375	190	125
IMF_h2	187	96	62.6
IMF_h3	93	46	31.3
IMF_h4	61	30	20.7
IMF_h5	29	15	9.99
IMF_residual	0	0	0.333

Orthogonality Matrix

Channel	IMF_h1	IMF_...	IMF_...	IMF_...	IMF_h5	IMF_residual
IMF_h1	1	0.294	-0.0295	-0.133	0.0305	0.00669
IMF_h2	0.294	1	0.331	0.0987	0.0234	-0.0413
IMF_h3	-0.0295	0.331	1	0.204	0.0144	0.0149
IMF_h4	-0.133	0.0987	0.204	1	0.0408	-0.0618
IMF_h5	0.0305	0.0234	0.0144	0.0408	1	-0.029
IMF_residual	0.00669	-0.0413	0.0149	-0.0618	-0.029	1

Percentage Power

Chanel	Power (%)
IMF_h1	0.000267
IMF_h2	0.000875
IMF_h3	0.0122
IMF_h4	0.358
IMF_h5	99.6

There are two options and a refresh button at the top of the Reporter Window. The first option is the way for displaying the decimal numbers. The default is *General* decimal display. In the drop-down menu, there are also options of *Fixed* decimal display and the *Scientific* decimal display. The second option is to set how many decimal places to be displayed and the default is 3. Click on the *Refresh* button when either of the options has been modified and the calculations results in the table are updated with the new settings.

In this example, it shows that the residual is the original Exp function while Sine function is IMF5. The other IMFs are signals generated in the EEMD decomposition. However, due to their very low Power, they do not cause difficulties for evaluation.

Related Functions

EEMD, Channel Viewer.

3.7.3 NASA HHT-DPS(Professional Only)

NASA HHT Module based on NASA HHT-DPS is integrated and developed by DynaDx Corporation.

Typical uses for the NASA HHT Module include:

Time-frequency Analysis

Non-stationary data analysis

Noise filtering

Feature Extraction

...and many more...

Applications of NASA HHT Module include:

Earth Science,

Biomedical Science,

Civil Engineering,

Mechanical Engineering,

Electrical Engineering,

Finance,

and others….

The Hilbert-Huang Transform Data Processing System (HHT-DPS) is original a signal processing toolkit. It implements the Hilbert-Huang Transform (HHT), a modern method for signal and data analysis. It was developed by the National Aeronautics and Space Administration, at Goddard Space Flight Center.

NASA HHT Module Documentation

This documentation includes two segments, one is the algorithm descriptions of HHT, and the other is manipulation. The algorithm descriptions of NASA HHT follow HHT-DPS documentation.

HHT-DPS documentation draws upon information from the following sources (see patents section for additional sources):

Bendat, Julius S., Peirsol, Allen G. “Random Data Analysis and Measurement Procedures .” New York: Wiley-Interscience, 1986

Blank, K. “Hilbert-Huang Data Processing System” PIP Report, June 9, 2003

Cohen, Leon. Time-Frequency Analysis. New Jersey: Prentice Hall PTR, 1995

Hahn, Stefan L. Hilbert Transforms in Signal Processing. Norwood, Massachusetts: Artech House Inc., 1996

Huang, Norden E., et al. “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis,” Proceedings of the Royal Society, vol 454, pp.903-995, London: 1998

Huang, Norden E., et al. “A confidence limit for the empirical mode decomposition and Hilbert spectral analysis,” Proceedings of the Royal Society, vol 459, pp 2317-2345, London: 2003

Kizhner, Semion, et al. “On the Hilbert-Huang Transform Data Processing System Development”, Proceedings of the IEEE Aerospace Conference. vol 3, pp 1979, 2004

Poularikas, Alexander D. ed. The Transforms and Applications Handbook. Boca Raton, FL: CRC Press, 1995

Weisstein, Eric W. "Discontinuity." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Discontinuity.html

Wu, Zhaohua and Huang, Norden E. “A study of the characteristics of white noise using the empirical mode decomposition method”, Proceedings of the Royal Society, vol 460, pp1597-1611, London 2004

HHT-DPS Documentation Authoring

This document was written and reviewed by Karin Blank, Norden Huang, Semion Kizhner, Per Gloersen, Tom Flatley, David Petrick, for the HHT-DPS version 1.4.

Copyright and Licensing

Hilbert-Huang Data Processing System Copyright

Copyright United States Government as represented by the Administrator of the National Aeronautics and Space Administration

KissFFT

This software includes KissFFT which is released under the following license: Copyright (c) 2003-2004 Mark Borgerding

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this

list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the author nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Licensing Authorization

DynaDx Corporation and NASA have signed an IP license agreement, effective on July 22, 2009. The agreement includes 11 patents of Hilbert-Huang Transformation (HHT) technology and its related applications. DynaDx has exclusive license right on these IPs.

Introduction

What is the HHT?

The Hilbert-Huang Transform (HHT) method is composed of multiple algorithms intended to filter and analyze the data.

These algorithms include:

-Empirical Mode Decomposition, where data is broken down into Intrinsic Mode Functions (IMFs)

-Normalized Hilbert Transform, which converts the IMFs to the time-frequency domain

-Generalized Zero Crossing, an alternative method for calculating local frequency from IMFs

-Degree of Stationary, a method for ascertaining the amount of variation in a signal

NASA HHT Module

NASA HHT Module has multiple modules base on above-mentioned algorithms.
NASA HHT Module includes:

NASA EMD, which decomposes the signal into many Intrinsic Mode Functions (IMFs)

NASA Hilbert Transform, which converts the IMFs to the time-frequency domain by Normalized Hilbert Transform.

NASA GZC, using Generalized Zero Crossing method to calculate zero-crossing rate from IMFs.

NASA Hilbert Spectrum, converts all IMFs into spectrogram by Normalized Hilbert Transform.

NASA GZC Spectrum, converts all IMFs into spectrogram by Generalized Zero Crossing method.

NASA Degree of Stationary, which calculates the variation degree of s the ignal.

3.7.3.1 NASA EMD

Introduction

Empirical Mode Decomposition (EMD) is a method of decomposing data into Intrinsic Mode Functions (IMF). Unlike many traditional filters, which are based in the frequency domain, the EMD is a time-domain based decomposition and does not make assumptions on the stationary and linear properties of the signal. By eliminating these assumptions and using the Hilbert Transform to obtain the instantaneous frequency, sharper resolution in the frequency domain can be obtained compared to conventional methods.

In order to use the Hilbert Transform to obtain meaningful results, it is mathematically necessary to have data that are locally symmetrical with respect to the y axis' zero-line. In such data, the rising side of a wave is relatively similar to its descending side, and vise versa. Applying the Hilbert Transform to unsymmetrical data yields several paradoxes that were defined by Cohen.

To avoid these issues, the concept of an Intrinsic Mode Function (IMF) was developed, which satisfies the necessary condition for obtaining the meaningful values of instantaneous frequency. The technical definition of an IMF is a signal in which the number of local extrema and the number of zero crossings should be the same, or at most differ only by one. This rule, implemented with the EMD, produces IMFs with well-defined instantaneous frequency.

Algorithms

Empirical Mode Decomposition Algorithm

The EMD process can be divided into several stages, which consist of the following specific steps:

Finding local extrema -Local extrema are sections of data that can be considered a "local maximum" or "local minimum". They are the points where the slope on one side of the point has a different sign than that of the slope on the other side.

Creating an envelope -Envelopes are generated by connecting a series of points with a smooth curve; in this case, the cubic spline interpolation is used. Here, envelopes come in two varieties: an "upper envelope" in which the local maximum extrema are interpolated, and a "lower envelope" where the local minimum extrema are interpolated. Ideally, the envelopes should encompass, but do not cut into the data.

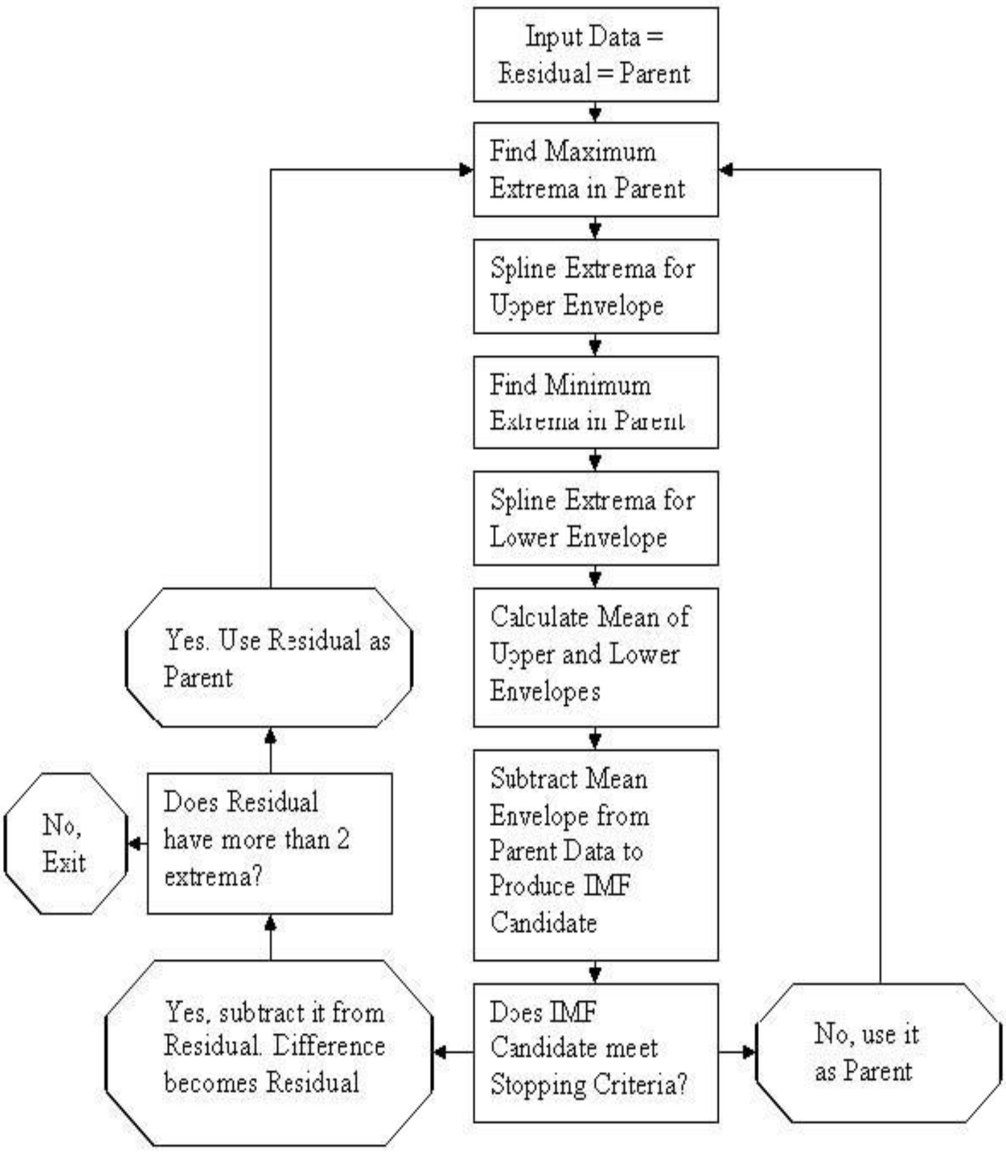
Assembling a local mean -This mean is created by averaging, point by point, the upper and lower envelopes.

Creating an IMF candidate -This is done by subtracting the local mean from the original signal. The result is an "IMF candidate" - it may or may not actually be an IMF.

Stopping Criteria -These rules are used to test if an IMF candidate can be considered as an IMF. This consists of two checks - first to see if the signal meets the definition of an IMF, and secondly, if the signal has been sifted enough times.
Sift -Completion of a set of all the steps above, regardless if an IMF is found or not, is called a "sift".

Creating the residual -If the stopping criteria are satisfied, the IMF is subtracted from the original data to create an interim residual.Then,this residual is put through a subsequent sifting process.

These combine an EMD Flow Chart EMD Sift Step-by-Step as follows:

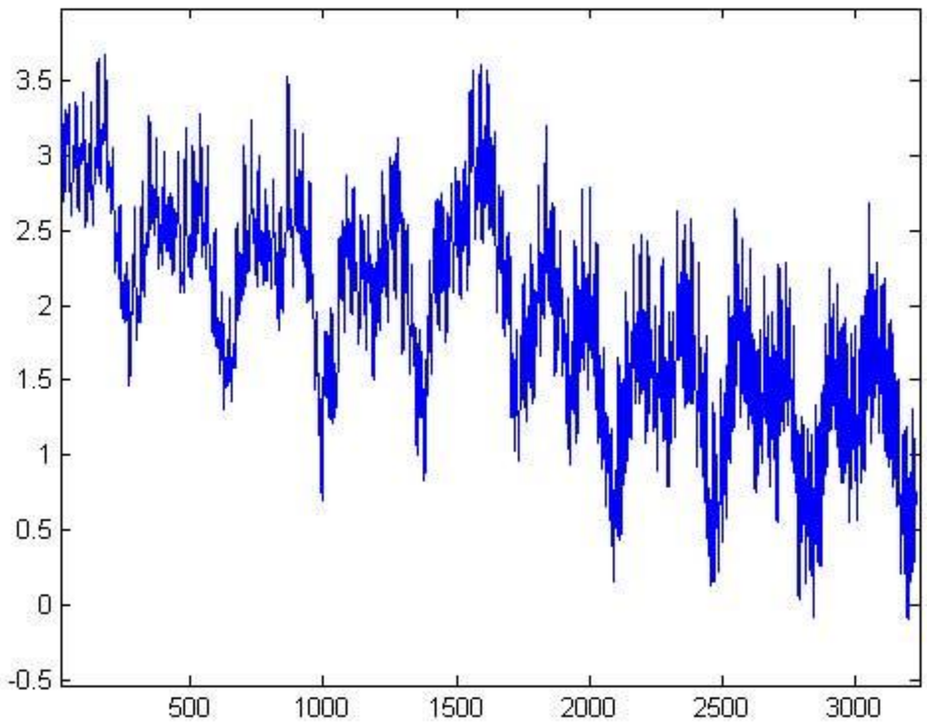


To understand the effects of the above process, the following is a brief illustration of

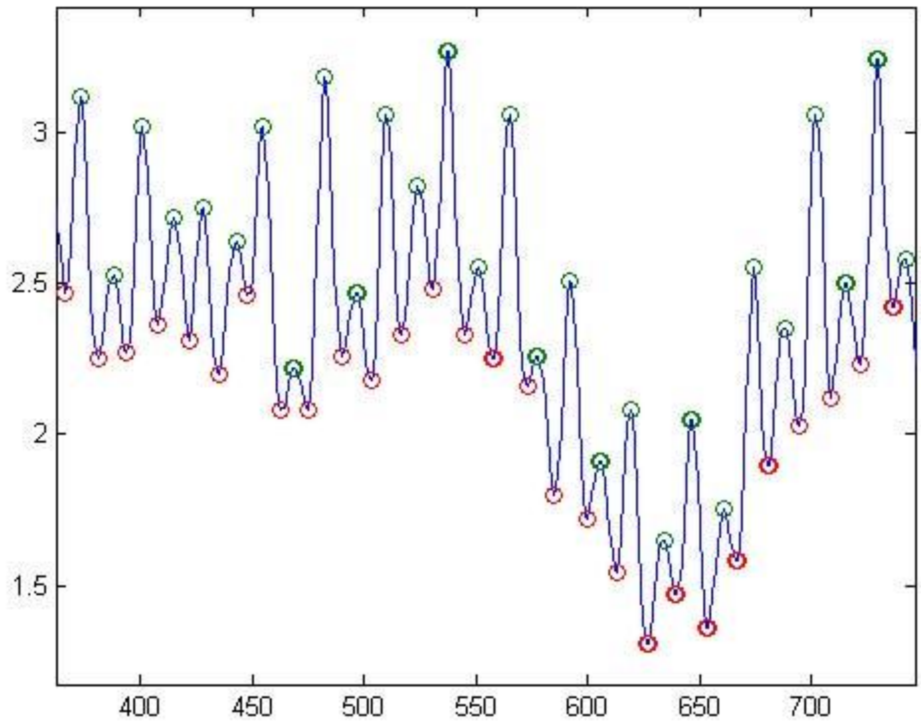
the steps taken for an entire sift.

First, the data is inputted into the EMD. This is the lod78.dat dataset that is included in the sample_data folder of the HHT-DPS. It contains the differences in the length of day over a period of ~9 years.

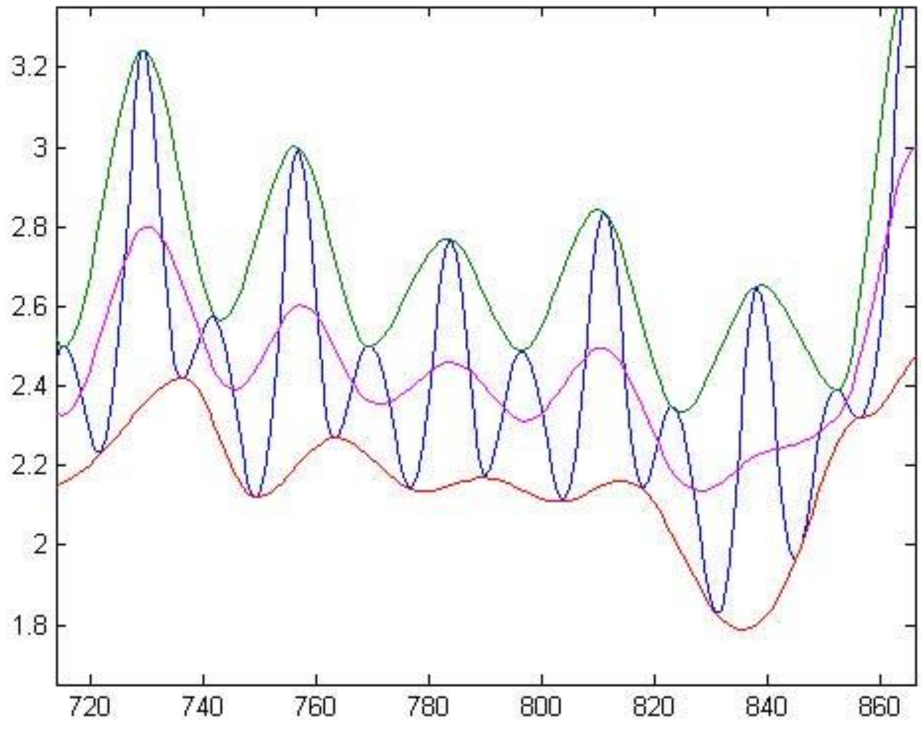
Length of Day data. The y axis is the difference in microseconds; x is the number of days since measurement began.



Zoom in of lod78 data. Original data is blue, the maximum extrema are green, and the minimum extrema are red.

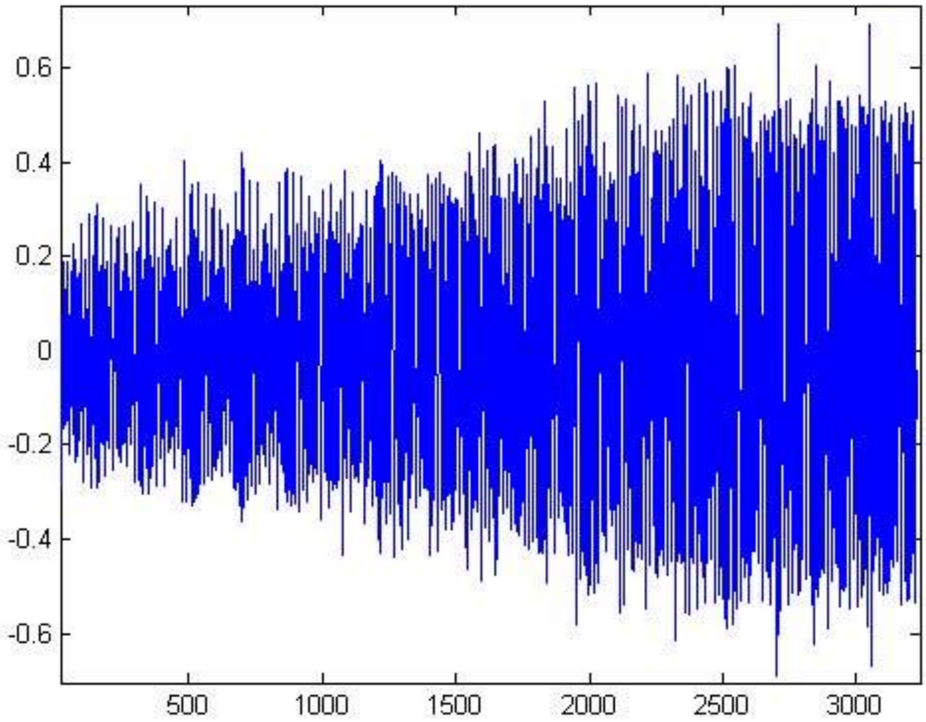


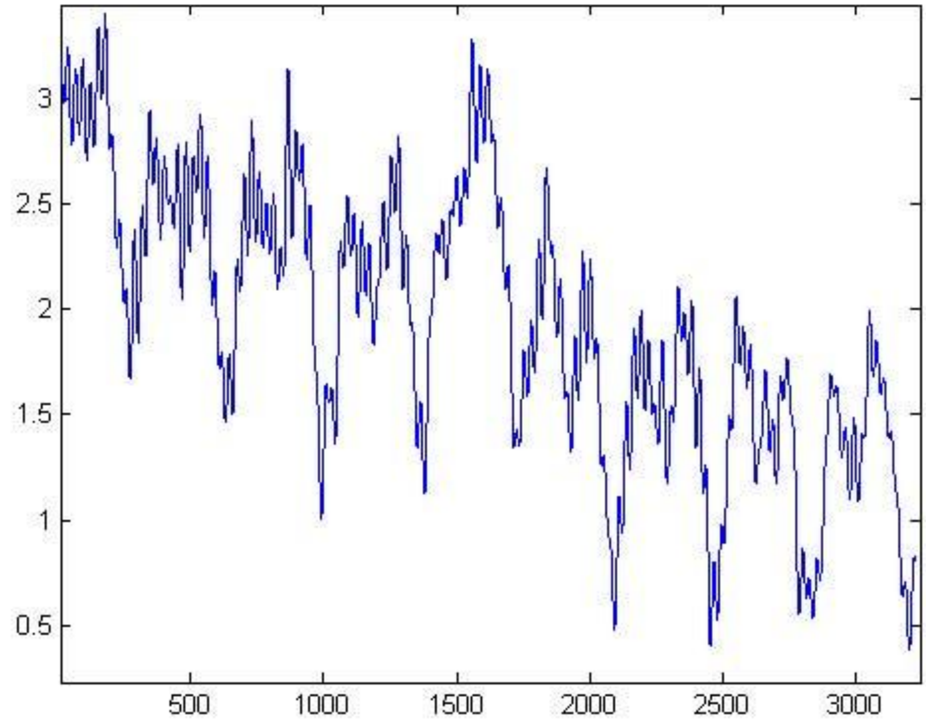
Zoom in of lod78 data. Original data is blue, the maximum envelope is green, the minimum envelope is red, and the local mean is purple.



The first IMF found from lod78

It is then subtracted from its originating signal, in this case lod78 (for others it would be the current residual), generating the residual signal.





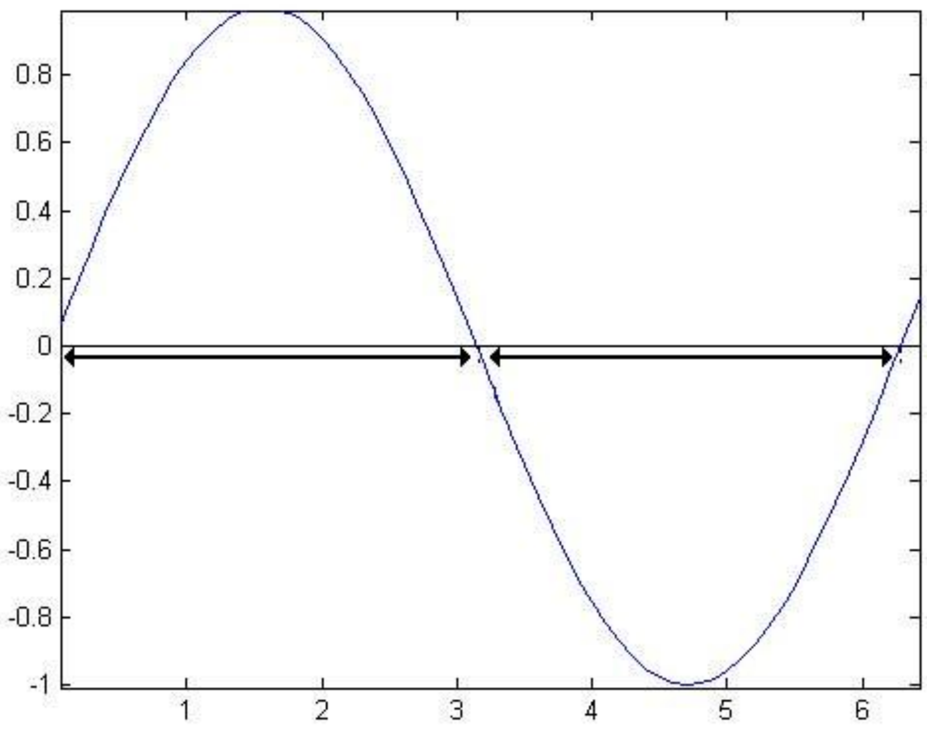
This residual is then put through the sifting process. The EMD continues generating signals until the residual signal contains less than two extrema. The process is then finished.

Intermittency

Intermittency is a method of removing certain frequencies from an IMF. This is done during the sifting process, directly after an IMF is found but before it has been subtracted from the residual. According to the EMD Flow Chart, this would occur between the steps titled "Does IMF candidate meet stopping criteria" and "Yes, subtract it from Residual. Difference becomes residual”

The algorithm for intermittency is as follows. From the IMF, distances between zero crossings are calculated. If the distance is greater than the user-defined threshold value, the signal between the two zero crossings are zeroed out. If not, that section of the signal is retained.

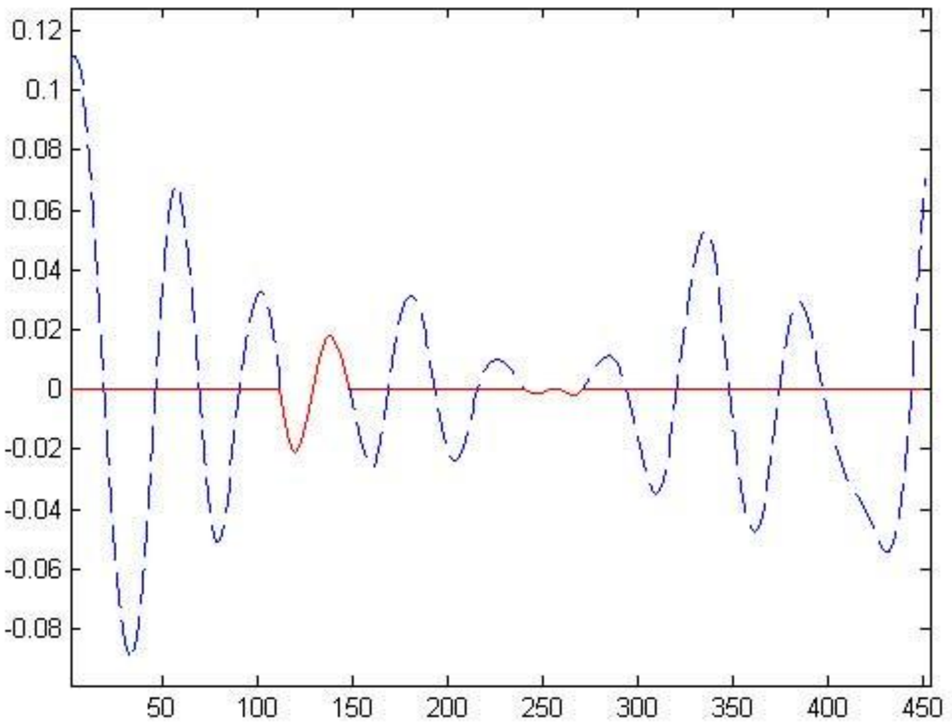
The distances between zero crossings, the value for which threshold is compared against in intermittency.

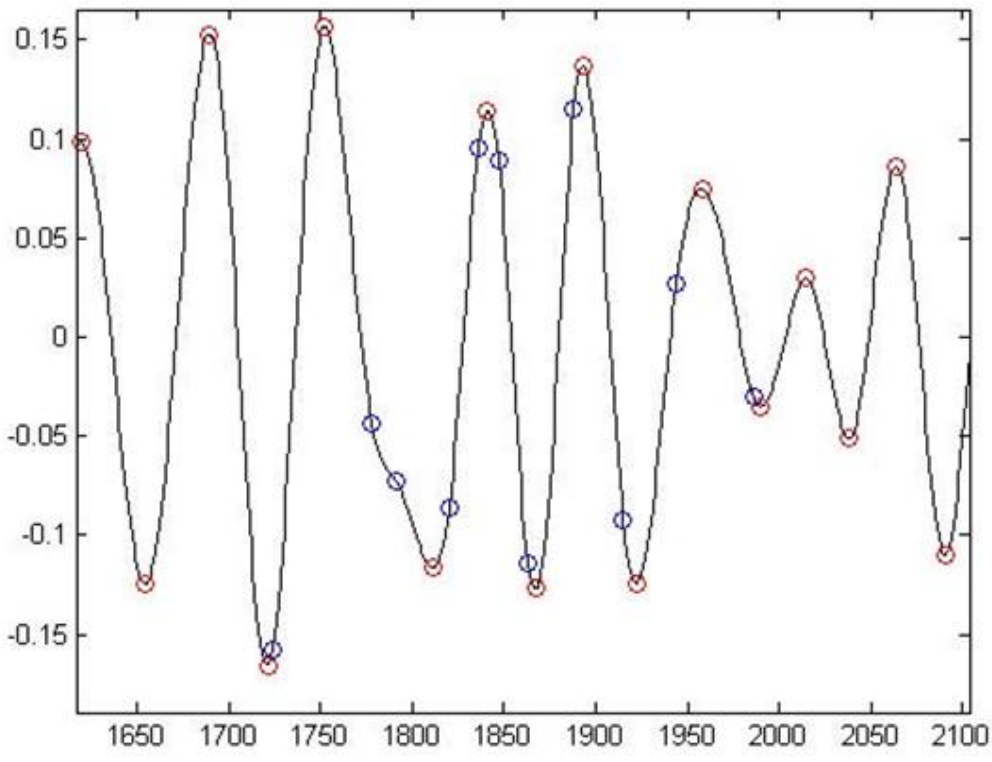


Intermittency is useful for collecting oscillations of similar scales to reside in one IMF, which will avoid mode mixing from an IMF.

The threshold is measured in the integer value of "number of points". It is not affected by the time offset or timescale of the signal.

Example of an IMF before and after an intermittency threshold of "20" is used. The blue dashed line is the IMF prior to intermittency. The red line is the IMF after intermittency.





Properties

This module accepts input of Signal (which could be real number, single channel, regular) and Audio (which could be real number, single channel, regular).The output is real number, multi-channel, Regular signal.

Option	Meaning	Default
Method	EMD method includes Standard, IntermittencyTest, and Ensemble.	Standard
PredictionType	The prediction type of endpoint, ex: PatternPrediction, CopyEndpoint, and Endpoint.	PatternPrediction
SiftCriteria	The number of times an IMF candidate must pass the test before considered an IMF.	3
MaxImfCount	Maximum number of IMFs to generate. Should be set high, this is safeguard for dataset where it may run seemingly indefinetly.	10
MaxSift	Maximum number of sift to generate IMF. It is intended mostly to prevent the EMD from getting stuck in an infinite loop.	10

IntermittencyTest

Option	Meaning	Default
Threshold	The intermittency threshold which unit is samples represent the distance between 2 zero crossing points. If the distances between zero crossings is greater than threshold, the two zero crossing would be zeroed out.	0

Ensemble

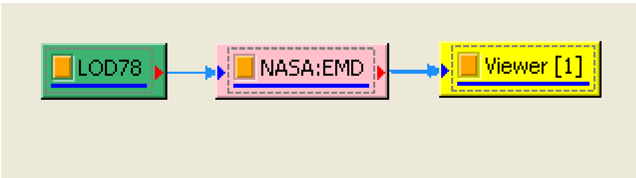
Option	Meaning	Default
EEMDNumber	Ensemble number for the EEMD.	20
EEMDEpsilon	The amplitude level of the noise (as a ratio to the standard deviation of input signal) added into input signal for ensemble.	0.1
NoiseCancellation	Specify if inverse noise was added for EMD process.	True
NoiseType	Added noise includes white noise or gaussian noise.	WhiteNoise

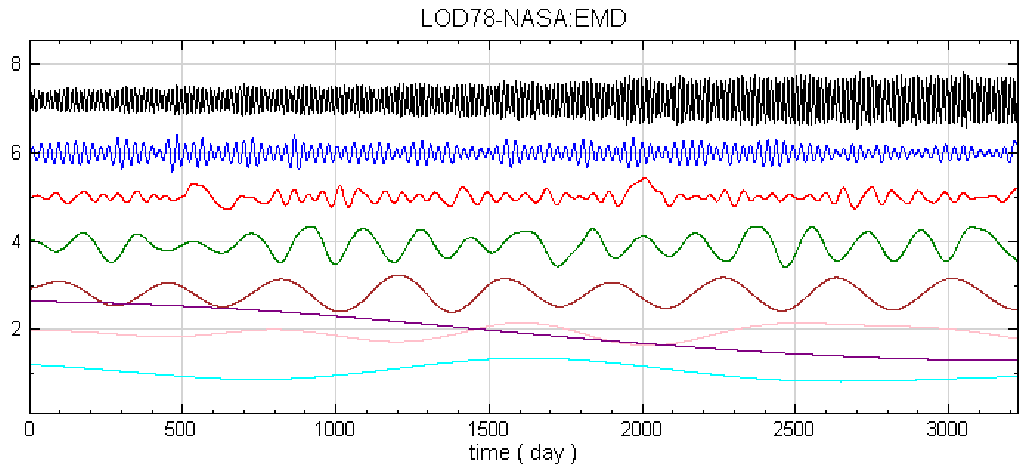
UserDefineSeed	If “True”, users could set the noise seed, “False”, seed is set according to current time.	True
Seed	The initial value of noise seed.	1
Sigma	The standard deviation of gaussian noise.	1

Example

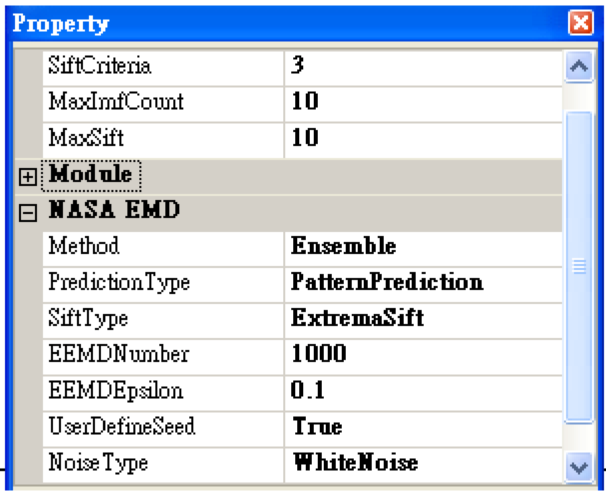
This example displays decomposition of LOD78 (Length of Day) by NASA EMD and decompose signal by the IntermittencyTest method of NASA EMD.

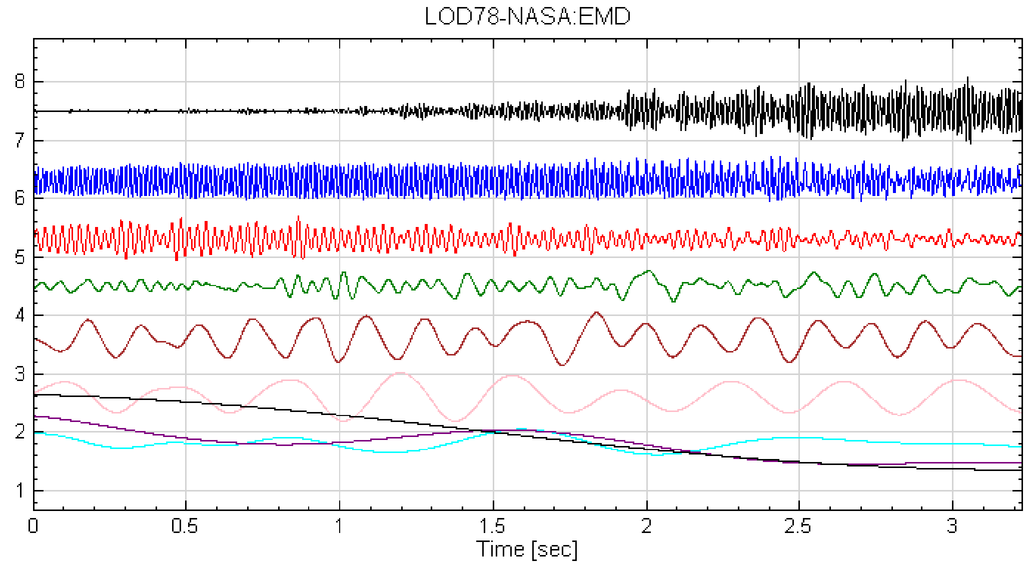
4. Use Source→Import data from file to read tfa file LOD78.tfa, in the installation directory (default to be C:\ProgramFiles\DynaDx\DataDemon\demo\HHT). Next perform Compute→NASA HHT→NASA EMD, and LOD78 would be decomposed into many IMFs.



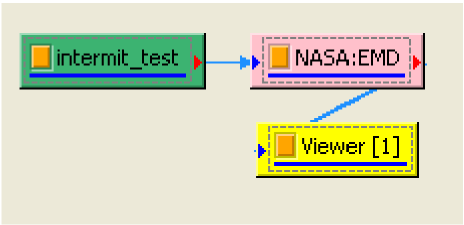


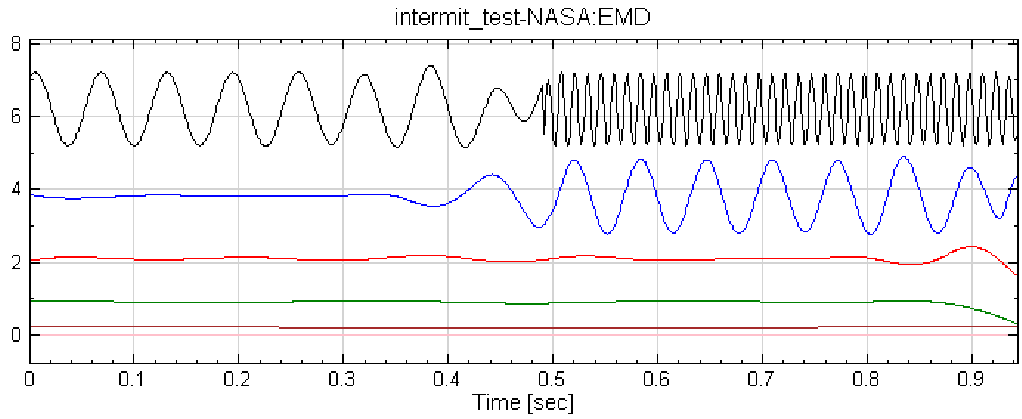
5. In NASA EMD, the property, set Method to Ensemble, and EEMDNumber to 1000. The result is displayed below.



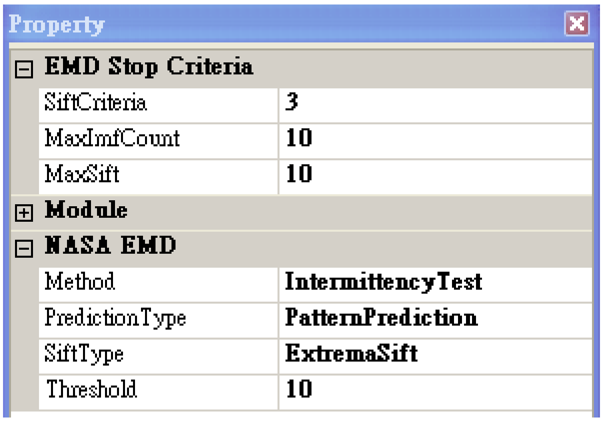


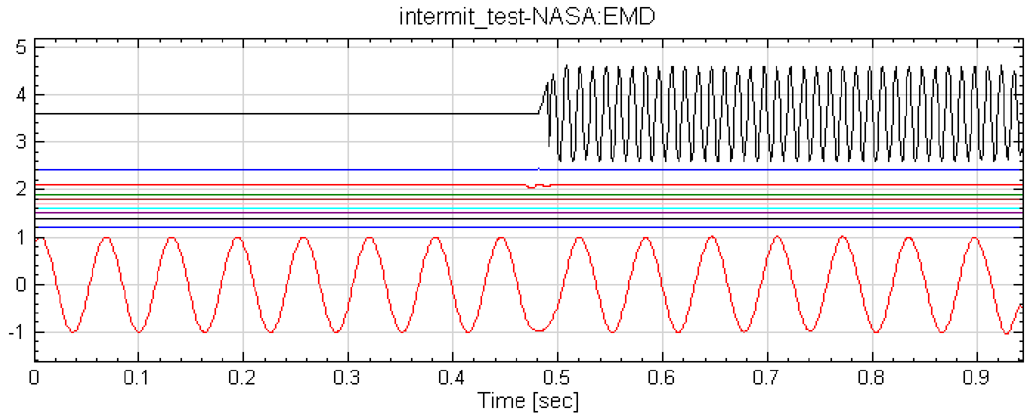
6. Reopen a new project, and Use Source→Import data from file to read tfa file intermit_test.tfa, in the installation directory (default to be C:\ProgramFiles\DynaDx\DataDemon\demo\HHT). Next perform Compute→NASA HHT→NASA EMD.





7. In NASA EMD, the property, set Method to IntermittencyTest, and Threshold to 10. The result is displayed below.





Related Functions

RCADA EEMD.

3.7.3.2 NASA Hilbert Transform

This module calculates instantaneous frequencies from the signal by Normalized Hilbert Transform. Hilbert Transform is described in the first segment, and then Normalzed Hilbert Transform is dicussed in the following segment.

Introduction

Hilbert Transform

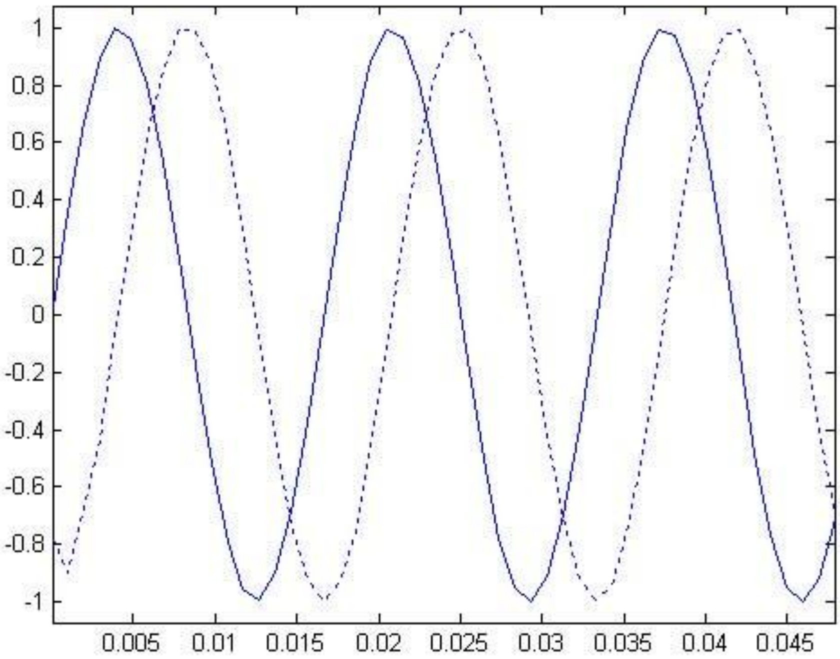
The Hilbert Transform is used to convert a one-dimensional real signal into an analytical signal. Unlike the Fourier transform, which changes a function of time into a function of frequency, the Hilbert transform gives the complex conjugate of the original data also in time domain.

There are many useful properties of the Hilbert transform, which include ninety-degree phase shifting, envelope function computations, and finding the instantaneous frequency. Our primary interest is for finding the instantaneous frequency. Although it may be tempting to apply it directly for signal analysis, few datasets processed would return meaningful information. This is due to four paradoxes identified by Cohen (see glossary, p 95).

$$H[x]=\frac{-1}{\pi}P\int_{-\infty}^{\infty}\frac{g(t)}{x-t}dt$$

The Hilbert Transform. P is the Cauchy principal value.

The Hilbert Transform produces a complex signal that consists of the original signal as the real part and the complex conjugate as the complex signal.



The analytic functions from applying the Hilbert Transform to sin(2 * Pi * 60 * x). Solid line is the real part of the transform; dotted line is the imaginary part of the transform. In case of simple sinusoidal functions, the Hilbert Transform will give a 90 Degree phase shift.

Once the Hilbert Transform is applied, instantaneous frequency can be obtained by considering a polar representation of the analytic signal. For this, the real part of the analytic signal is plotted on one axis; the imaginary part on the other. The instantaneous frequency is then derived by calculating the instantaneous angular change of the analytic signal.

$$\omega(t)=\frac{d}{dt}\tan^{-1}\frac{H_i}{H_r}$$

Equation for instantaneous angular frequency

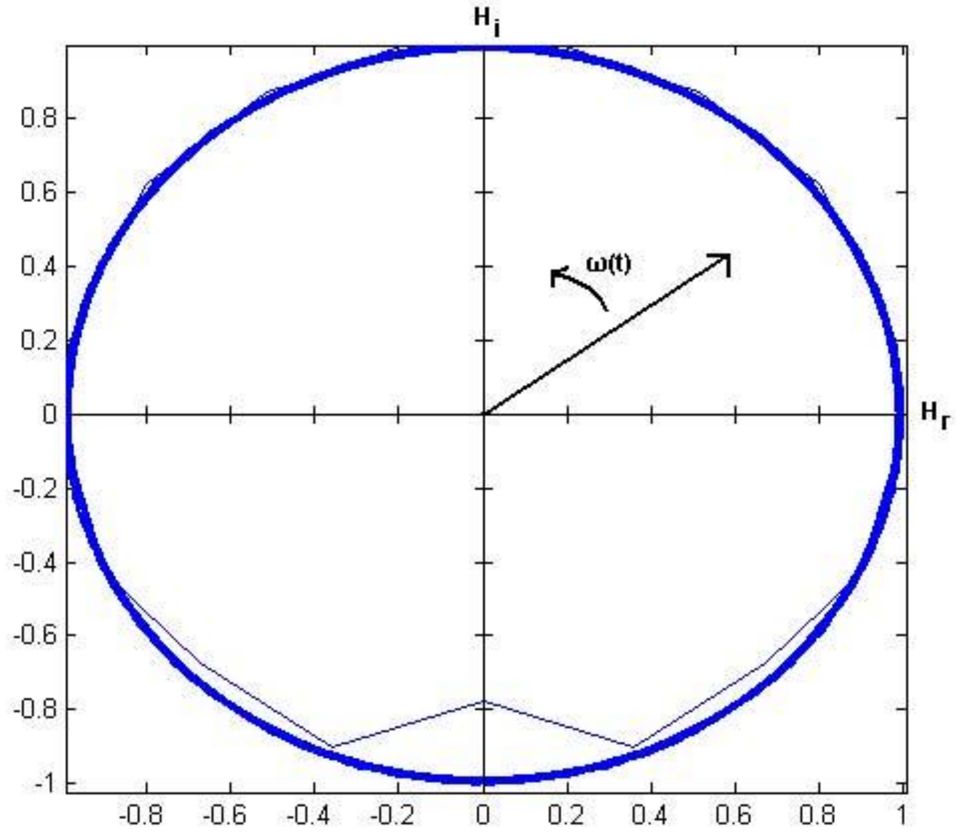


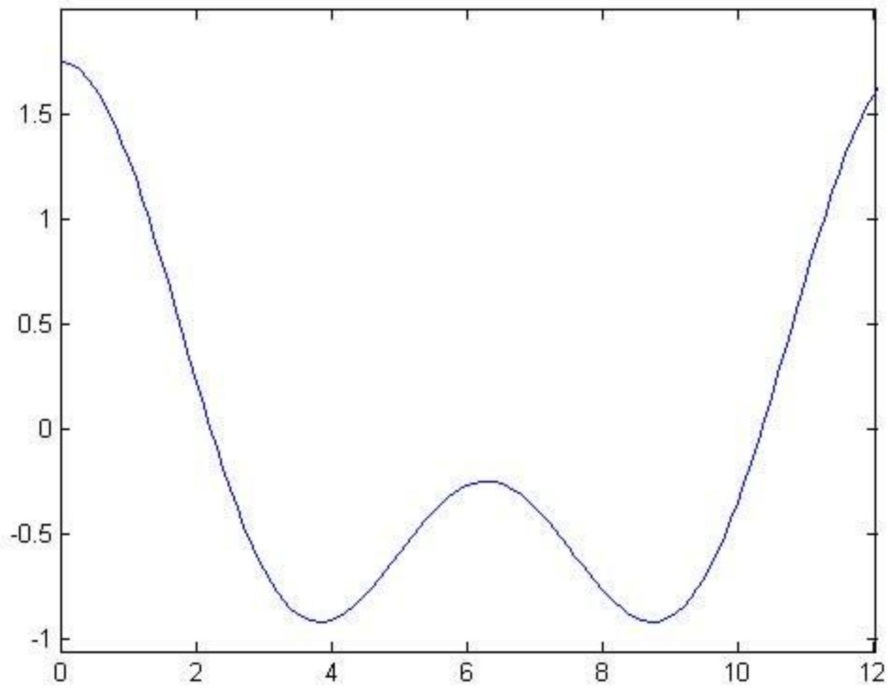
Image of the complex analytic signal of sin(2*Pi*60*x)

$$\Phi_c(t)=\frac{-t}{a^2+t^2}+j\frac{a}{a^2+t^2}$$

Equation for instantaneous frequency

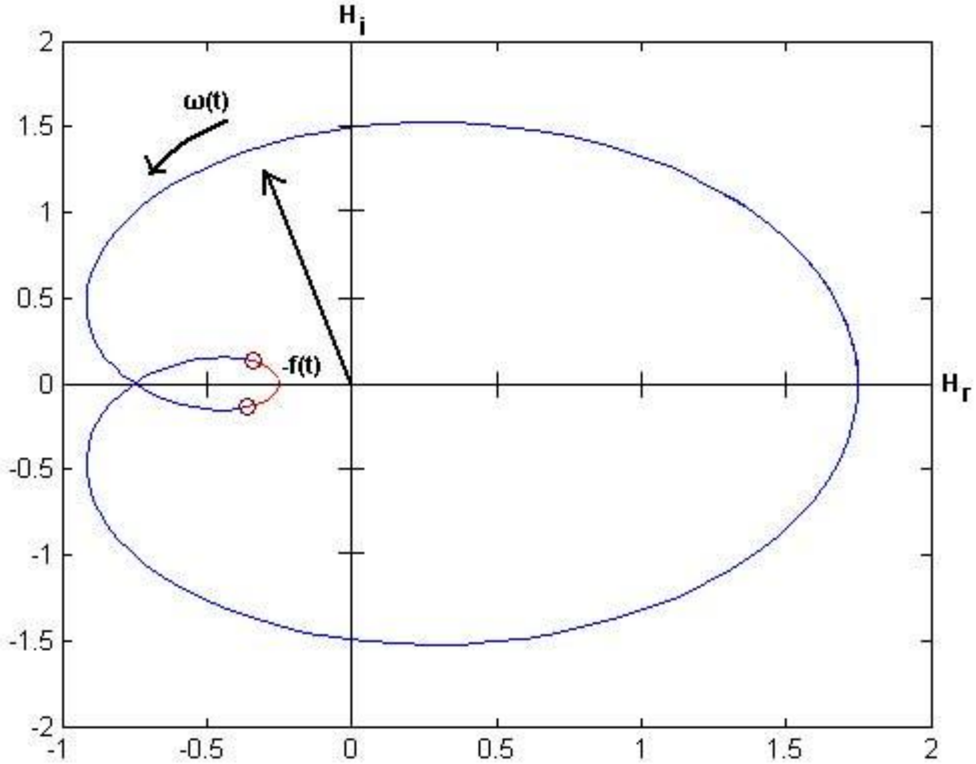
Negative Instantaneous Frequency

A problem with the Hilbert Transform, when used to obtain the instantaneous frequency, is that there are several situations where a signal will yield a negative frequency. Most of these situations are avoided by using the Empirical Mode Decomposition to eliminate riding waves (i.e. multiple extrema between zero-crossings). Intrinsic Mode Functions, however, only satisfy the necessary condition for a nonnegative frequency. This situation can be found, for example, in the signal y=cos(t/2) + .75*cos(t). This example can also be found in Hahn, 1996. Although this particular signal is not IMF, since the middle extrema does not cross the zero axis, it provides a clean example of the physical properties that can result in negative frequency.



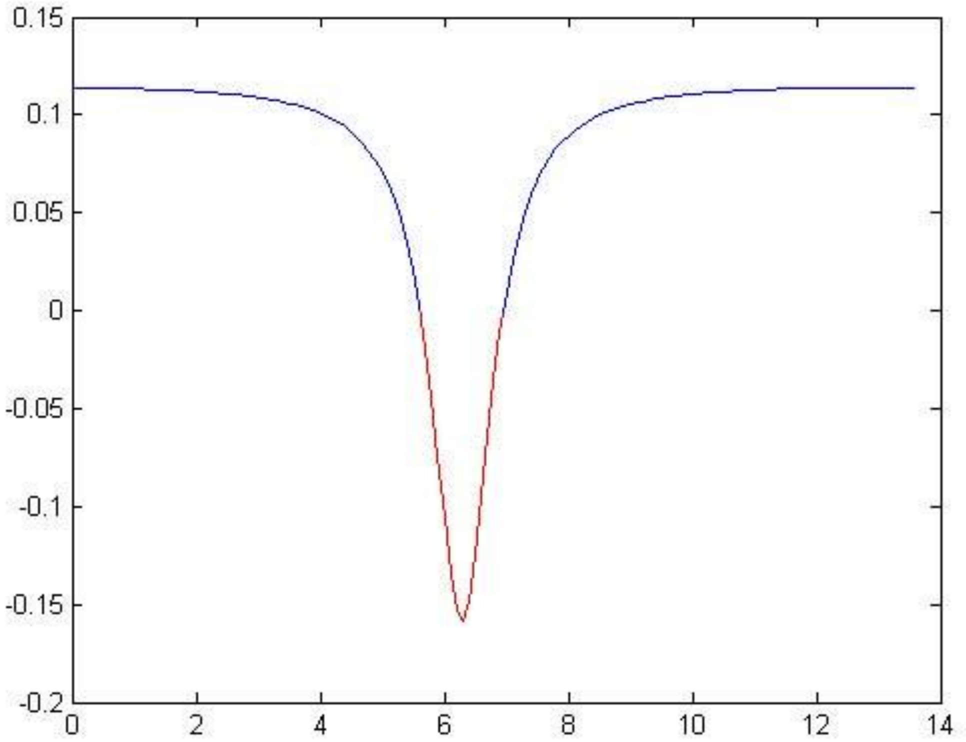
The signal, y=cos(t/2) + .75*cos(t). Negative frequency will occur due to the differences in amplitudes of the different waves.

Note the differences in amplitudes between the waves, Due to the phasor having to change the direction of rotation as the angular frequency is calculated, this will result in negative frequency. The limitation of the amplitude change is given by Bedrosian theorem (1963), which requires that the Fourier spectrum of the envelope should be disjoint with the Fourier spectrum of the carrier waves. Otherwise, the phase would not be separated from the amplitude fluctuation of the amplitude. Below is a graph of the analytic signal.



The analytic signal of y=cos(t/2) + .75*cos(t). The red sections indicate where negative frequency occurs due to the phasor's change of rotation

Since negative frequency does not have physical meaning in this situation, it is necessary to avoid generating it. In this particular instance, the solution is normalizing the signal.

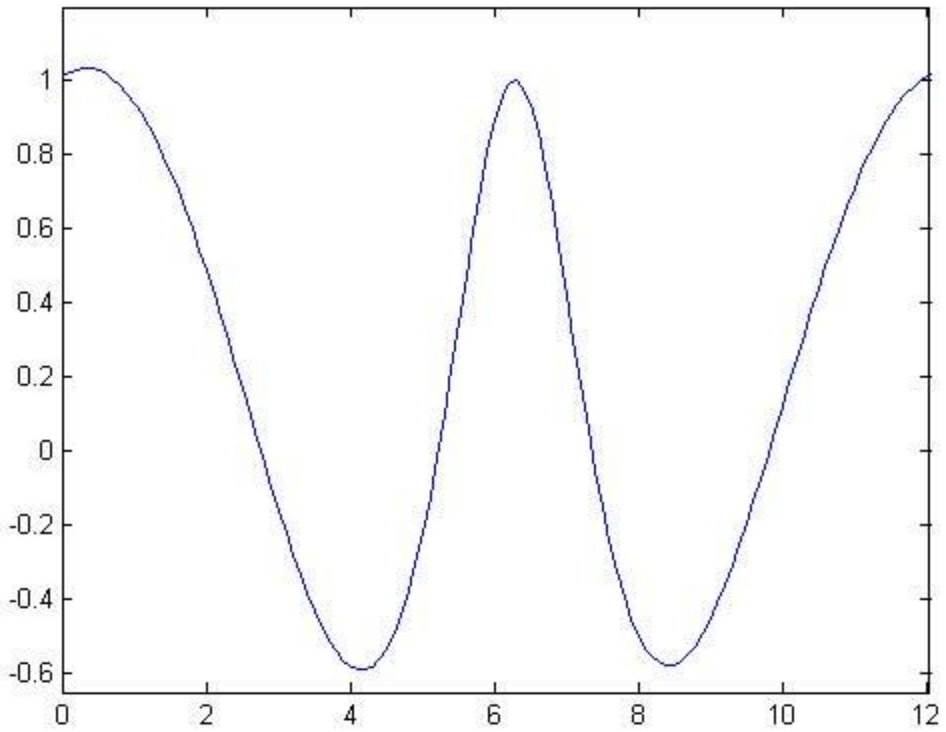


Instantaneous frequency of signal. Red indicates area where the result is negative frequency.

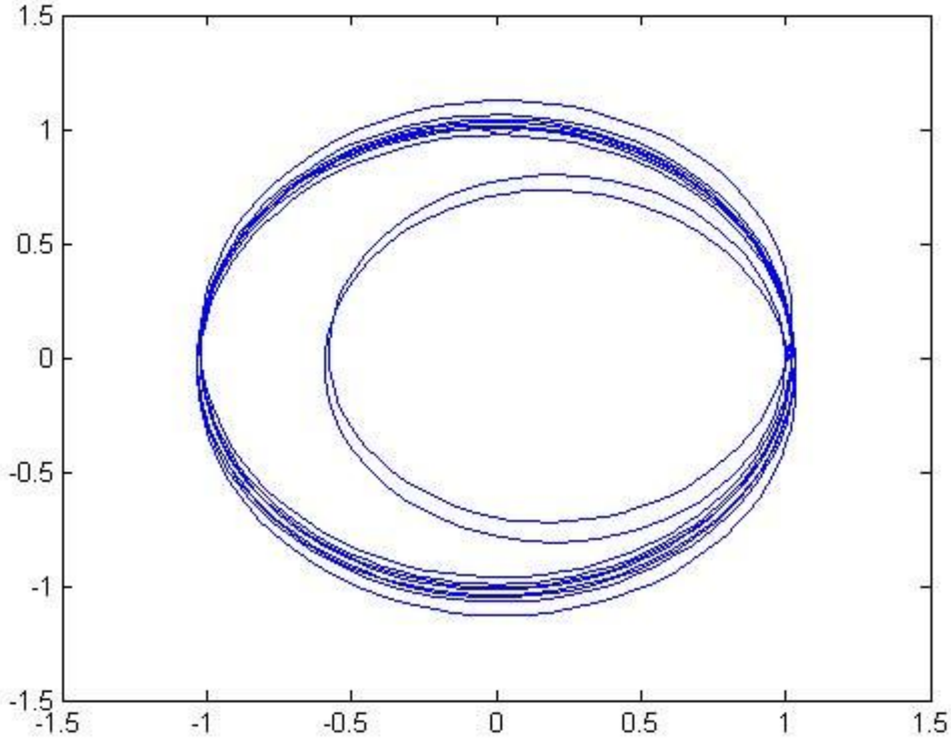
Normalized Hilbert Transform

To compensate for the effect of the different amplitudes of the waves, the HHT uses a preprocessing step on IMFs prior to applying the Hilbert Transform. Initially, the signal is normalized so that the maximum amplitude of each maximum extrema is exactly equal to one. Additionally, it also implements a step to neutralize the Gibbs effect by removing the discontinuity that often occurs at the signal's ends.

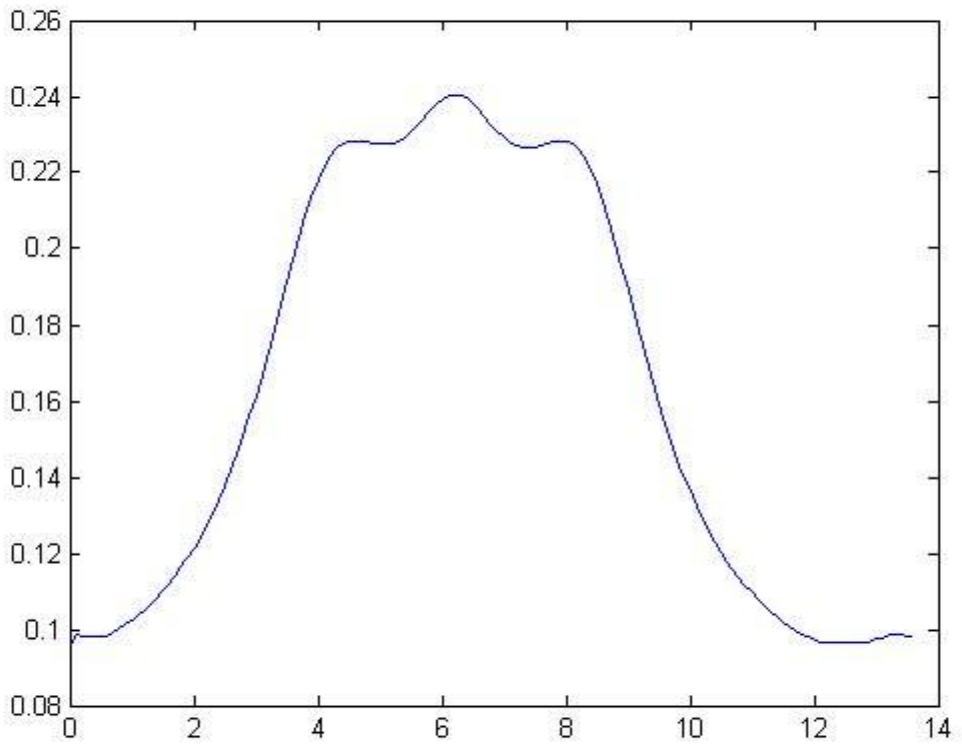
The following images demonstrate the process and its benefits.



y=cos(t/2) + .75*cos(t) after normalization



The analytic signal of the normalized y=cos(t/2) + .75*cos(t). The extra “loops” are due to sections of signal added to the ends of the data to help remove discontinuities.



The instantaneous frequency of the normalized signal.

Properties

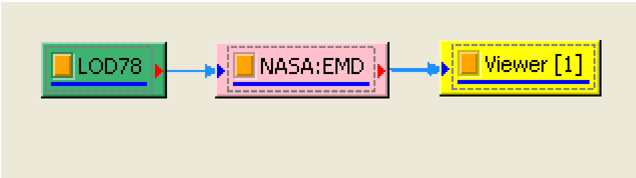
This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular).The output is real number, single channel or multi-channel, Regular signal.

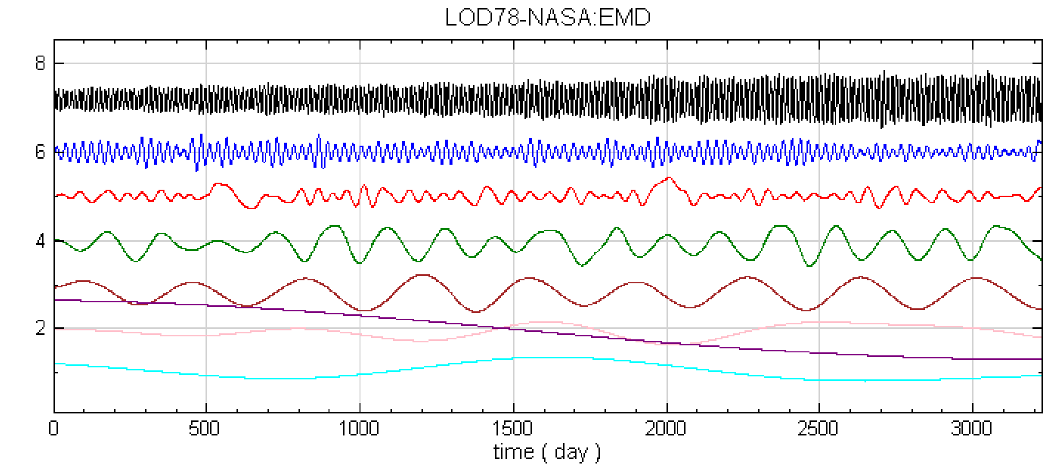
Option	Meaning	Default
SmoothPoint	Specify number of smooth point in median filter. A value of “0” indicates that no smooth should be used.	5

Example

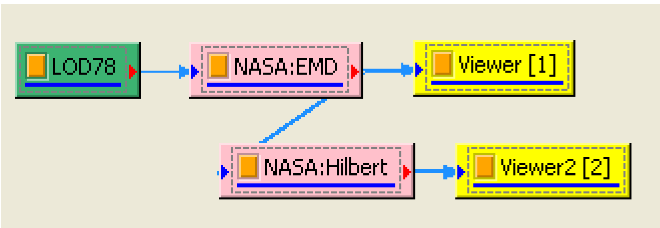
In this example, use this module to calculate instantaneous frequency form the EMD result of LOD78 (Length of Day). With time axis, frequency variation of each IMF could be observed.

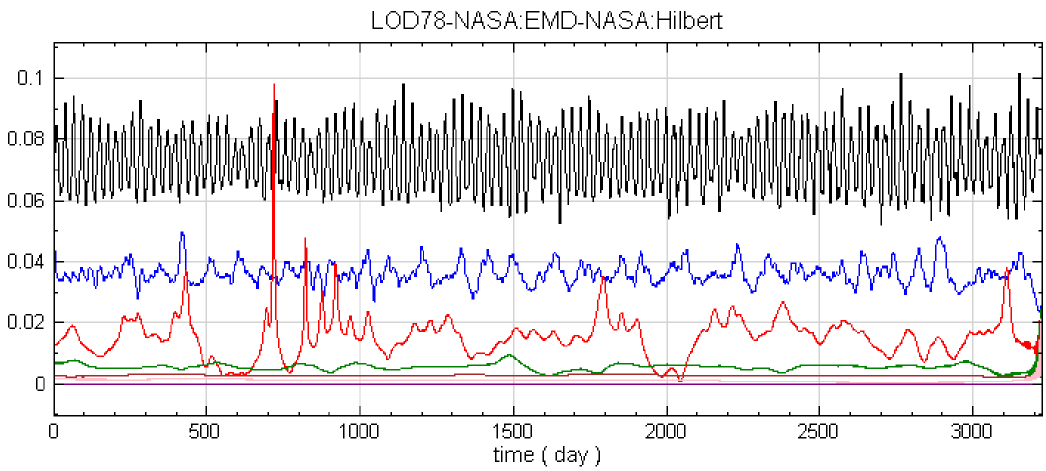
1. Use Source→Import data from file to read tfa file LOD78.tfa, in the installation directory (default to be C:\ProgramFiles\DynaDx\DataDemon\demo\HHT). Next perform Compute→NASA HHT→NASA EMD , and LOD78 would be decomposed into many IMFs.



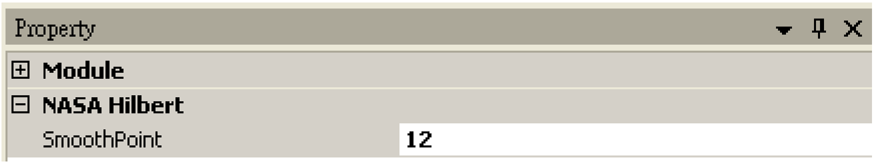


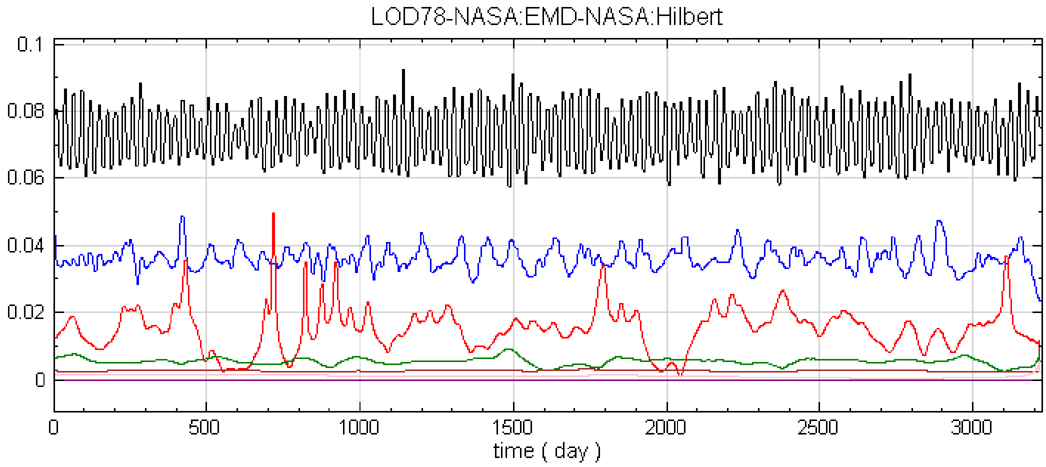
2. After NASA EMD, connect the module Compute→NASA HHT→NASA Hilbert Transform, whose property(SmoothPoint) is default value.





3. When SmoothPoint is changed to“12”, the instantaneous frequency would change smoothly with time.





Related Functions

Hilbert Transform, RCADA Instant. Frequency.

3.7.3.3 NASA GZC

This module whose full name is NASA Generalized Zero Crossing is an alternative module for calculating instantaneous frequency and instantaneous amplitude.

Introduction

Generalized Zero Crossing is the most direct method for calculating local frequency that is closely related to the original definition of frequency. This method only works with IMF-type signals. By using the physical properties contained within these signals, such as extrema, zero crossing, and the period.This new method can derive frequency directly in the time domain. Advantages of this method includes that it comes directly from the definition of frequency itself, as well as avoids issues with negative instantaneous frequency found in the Hilbert Transform. The disadvantage is that the local frequency is smoothed over at least a quarter of a wave length.

Algorithm

The Generalized Zero Crossing (GZC) method is a weighted calculation of the signal features local extrema, zero crossings, and the period. This method can only be successfully used on signals with the properties of an IMF, since these signals are not hampered with riding waves and other distortions. Each of these features alone can be used to obtain a frequency value of their own - for example, the length of the period, the traditional definition of frequency, but the GZC uses them all together for a distinct calculation that provides insight to local phenomenon.

$$\omega = \frac{1}{\Delta t_z}$$

Traditional definition of frequency, the inverse of the length of the period.

$$\bar{\omega} = \frac{1}{12}\left(\frac{1}{q} + \sum_{j=1}^{2} \frac{1}{h_j} + \sum_{k=1}^{4} \frac{1}{p_k}\right)$$

Frequency as defined by the Generalized Zero Crossing. p is the length of the period, h is the distance between zero crossings ("half" the period), and q is the distances between extrema and zero crossings ("quarter" of the period). These values are weighted to give advantage to measures that are more local. The wave period is defined by the critical points consisted of zero-crossings and local extrema. The “quarter”period is the time duration between neighboring critical points; the “half” period is the time duration between three consecutive neighboring points; and the “full”period is the time duration covering five consecutive neighboring critical points.

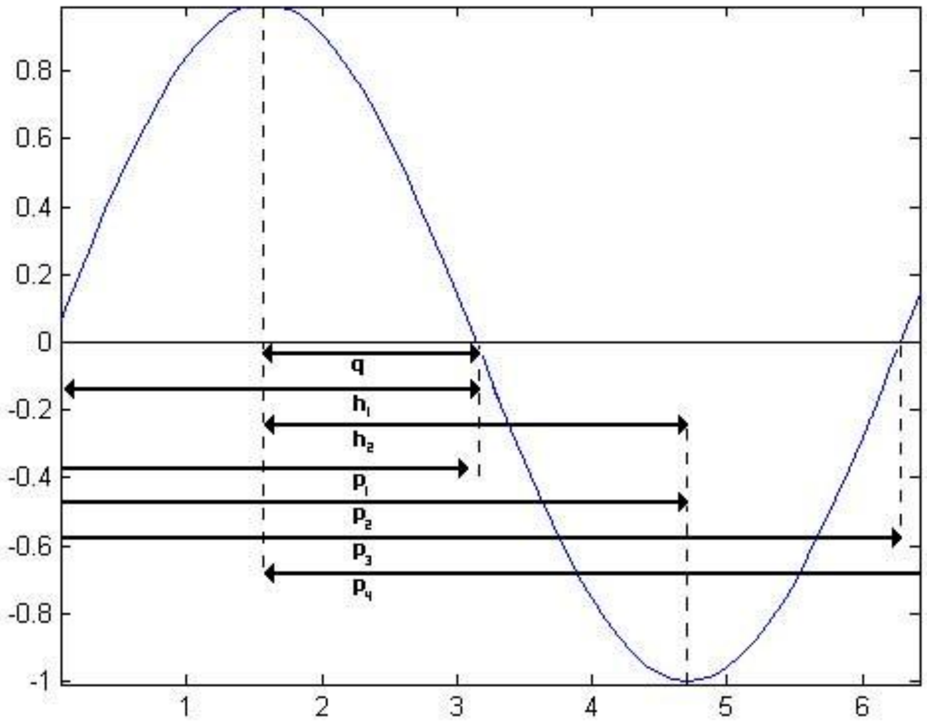


Image of a sine wave demonstrating the physical relationship between the above variables and the signal. The time frame of q is for which the frequency is being calculated.

An advantage of the GZC algorithm is that it can avoid negative frequency in calculating instantaneous frequency found in the Hilbert Transform. Also, since GZC relates directly to physical phenomenon in the data, it is clearer how more local frequency values can be obtained in relation to the traditional definition of frequency. The primary disadvantage GZC has over using the Hilbert Transform is that the resolution of the frequency values is limited to a quarter of the wavelength. Nevertheless, the clear physical meaning of the GZC made the frequency definition the true value in the mean.

Properties

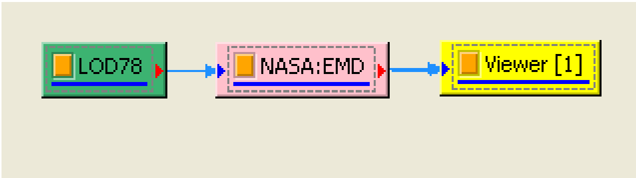
This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular).The output is real number, single channel or multi-channel, Regular signal.

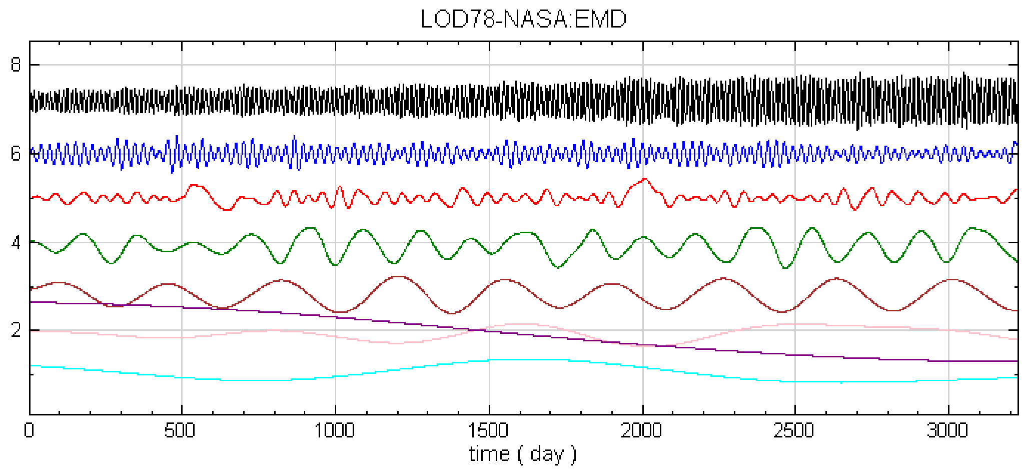
Option	Meaning	Default
OutputType	Specify type of output. Ex: InstantFreqency or InstantAmplitude.	InstantFreqency

Example

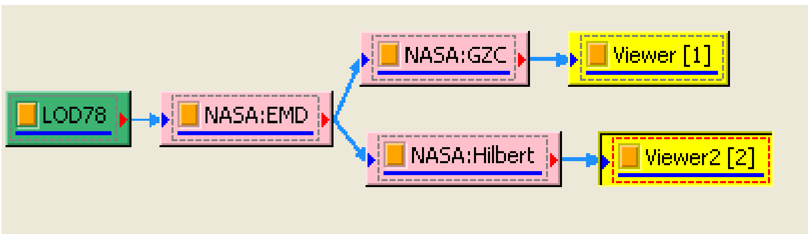
Compare with NASA Hilbert Transform, NASA GZC outputs instantaneous frequency or instantaneous amplitude. In this example, LOD78 is still decomposed and then all IMFs are transformed to frequency or amplitude with time.

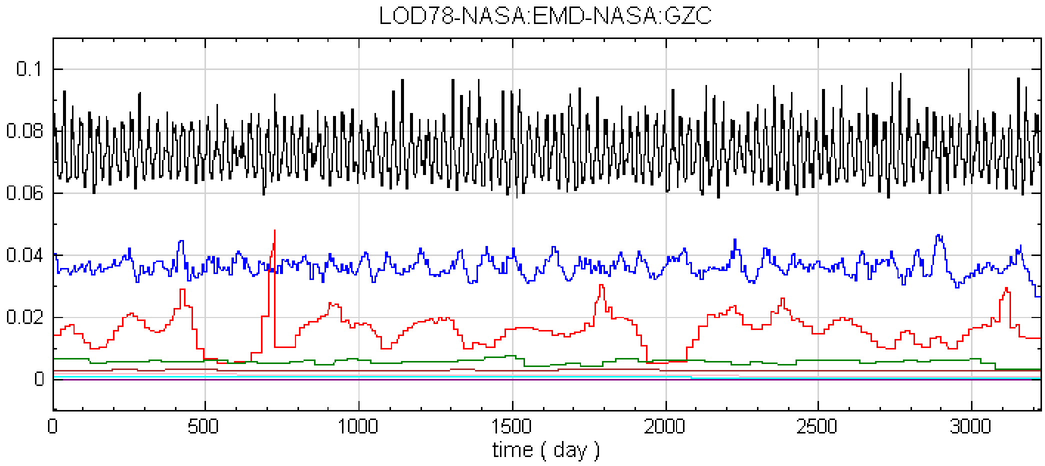
1. Still use Source→Import data from file to read tfa file LOD78.tfa, in the installation directory (default to be C:\ProgramFiles\DynaDx\DataDemon\demo\HHT). Next perform Compute→NASA HHT→NASA EMD , and LOD78 would be decomposed into many IMFs.

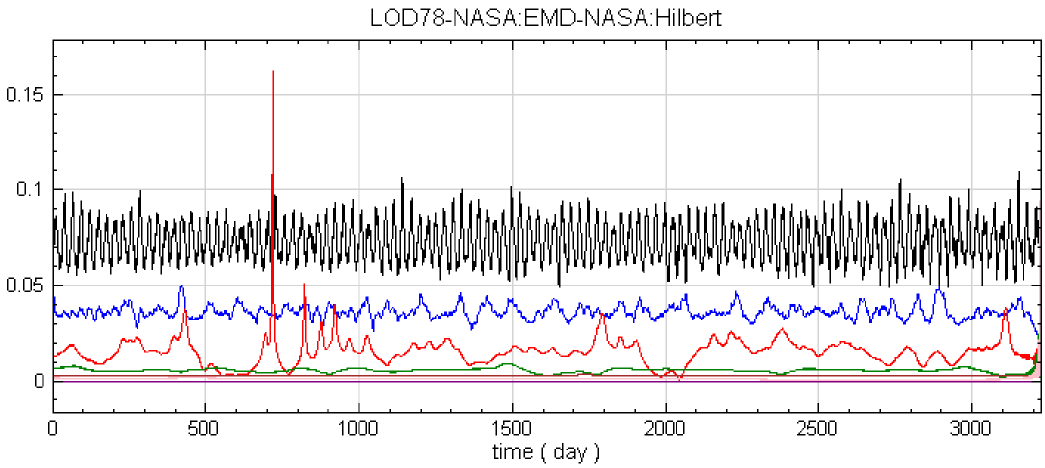




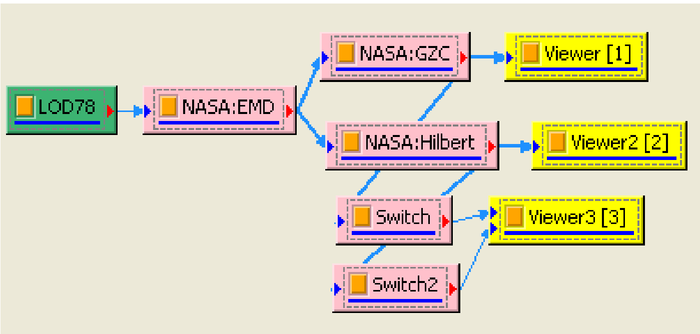
2. After NASA EMD, connect the module Compute→NASA HHT→NASA GZC, whose property(OutputType) is default value. In the same time, connect the module Compute → NASA HHT → NASA HilbertTransform whose property(SmootPoint) is set “1” to compare results from GZC results.

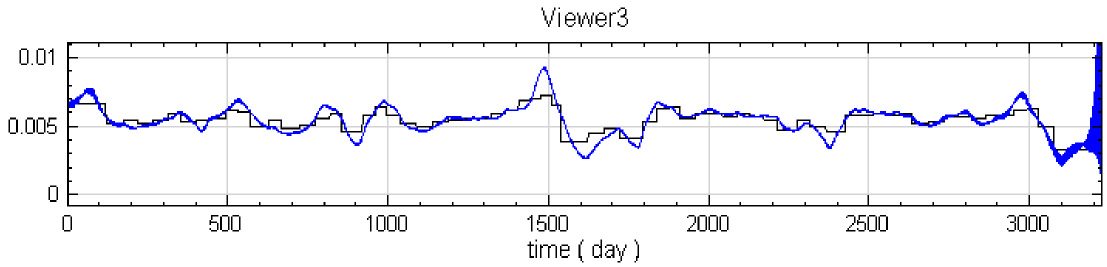






3. From NASA GZC and NASA HilbertTranfom, both connect the module Compute →Channel→Channel Switch, The Active Channel of SFOs is “4:CH4”, and after Channel Switch, connect Viewer→Channel Viewer to observe results. Result of HilbertTransform is smoother than result of GZC.





Related Functions

Hilbert Transform, RCADA Instant. Frequency.

3.7.3.4 NASA Hilbert Spectrum

NASA Hilbert Spectrum is to perform spectrogram via Normalized Hilbert Transform.

Introduction

Please see 3.7.5.2 NASA Hilbert Transform

Properties

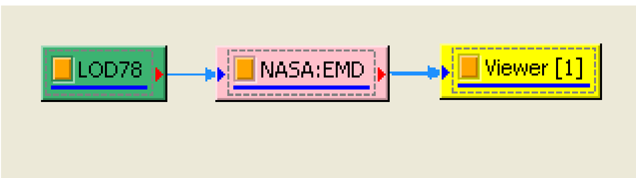
This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular).The output format is real number and signal-channel spectra data.

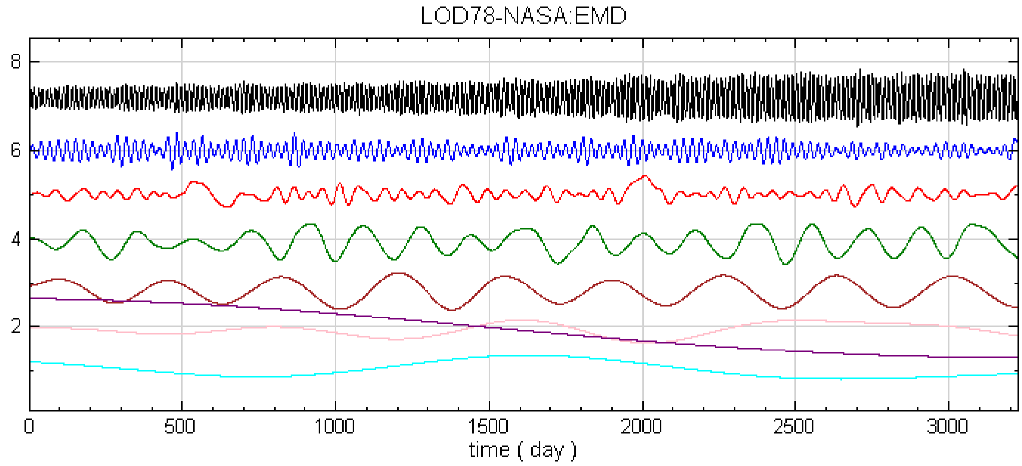
Option	Meaning	Default
FreqCount	The number of discrete lattice in frequency axis	256
TimeCount	The number of discrete lattice in time axis	1024
SmoothPoint	Sets number of median smooth points. A value of “0” indicates that smoothing would be not used.	5

Example

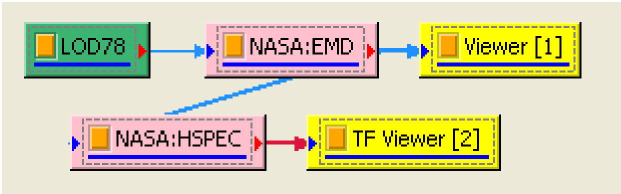
In this example, LOD78 is decomposed and then all IMFs are transformed into spectra by NASA Hilbert Spectrum.

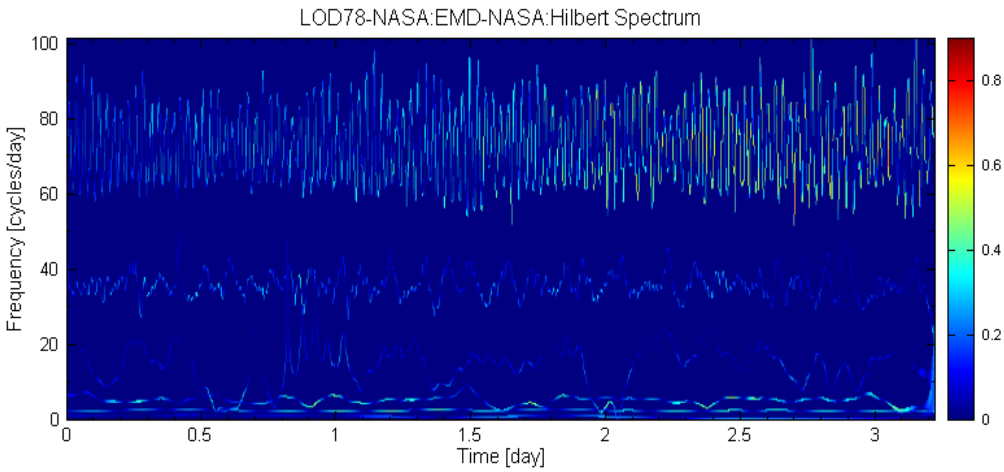
1. Use Source→Import data from file to read tfa file LOD78.tfa, in the installation directory (default to be C:\ProgramFiles\DynaDx\DataDemon\demo\HHT). Next perform Compute→NASA HHT→NASA EMD , and LOD78 would be decomposed into many IMFs.





2. Lastly use Compute→NASA HHT→NASA Hilbert Spectrum to connect NASA EMD, and furthermore time-frequency result is displayed in the TFA Viewer.





Related Functions

Hilbert Spectrum, RCADA Spectrum

3.7.3.5 NASA GZC Spectrum

NASA GZCSpectrum is to perform spectrogram via Generalized Zero Crossing method.

Introduction

Please see 3.7.5.3 NASA GZC

Properties

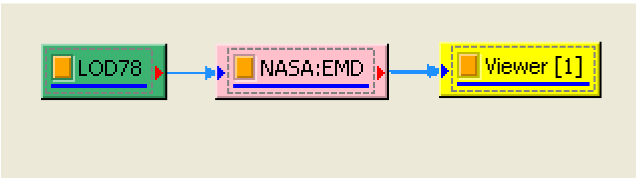
This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular) and Audio (which could be real number, single channel or multi-channel, Regular).The output format is real number and signal-channel spectra data.

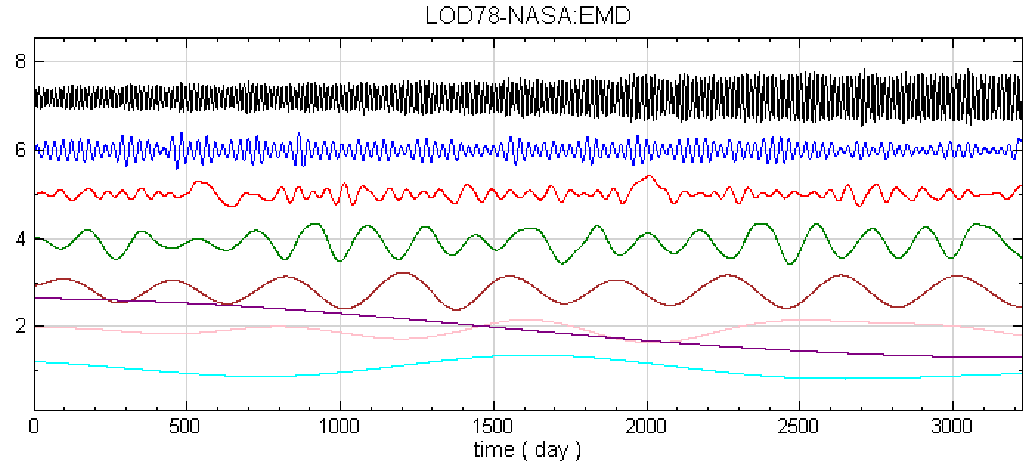
Option	Meaning	Default
FreqCount	The number of discrete lattice in frequency axis	256
TimeCount	The number of discrete lattice in time axis	1024

Example

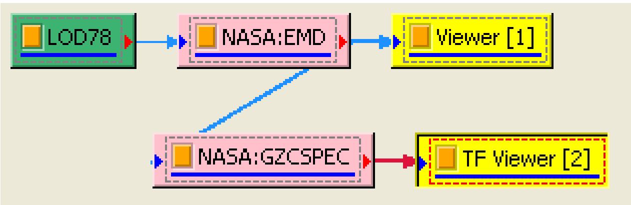
In this example, LOD78 is decomposed and then all IMFs are transformed into spectra by NASA GZC Spectrum.

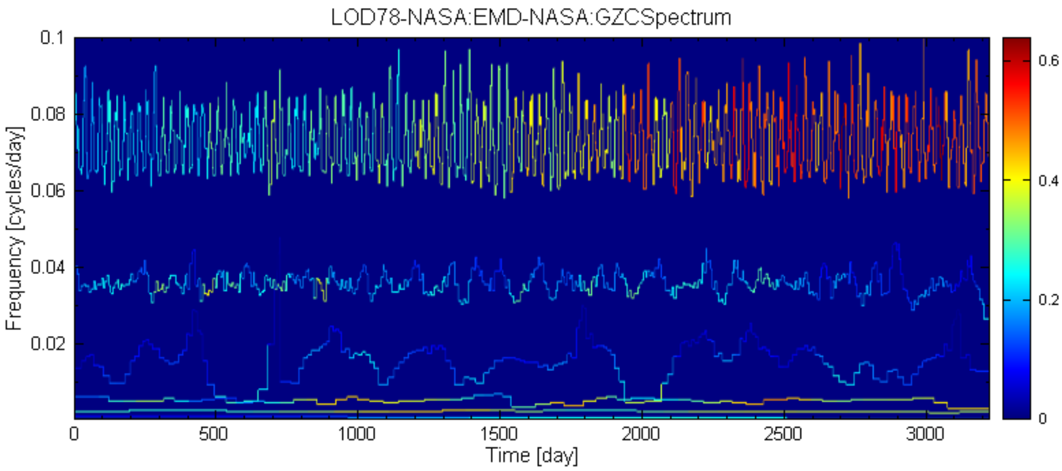
1. use Source→Import data from file to read tfa file LOD78.tfa, in the installation directory (default to be C:\ProgramFiles\DynaDx\DataDemon\demo\HHT). Next perform Compute→NASA HHT→NASA EMD , and LOD78 would be decomposed into many IMFs.





2. Then use Compute→NASA HHT→NASA GZCSpectrum to connect NASA EMD, and furthermore time-frequency result is displayed in the TFA Viewer.





Related Functions

Hilbert Spectrum, RCADA Spectrum

3.7.3.6 NASA Degree of Stationary

This module quantifies variety of each frequency form spectrogram. The algorithm of this module is implemented according to definition of Degree of Statistic Stationary (Norden E. Huang et al 1998), and NASA HHT-DPS named it as Degree of stationary.

Introduction

For a time-frequency distribution, $H(\omega, t)$

Define the marginal frequency as $n(\omega) = \frac{1}{T}\int_T H(\omega, t)dt$;

Then Degree of Stationary is computed as follows.

$$DS(\omega) = \frac{1}{T}\int_0^T [1 - \frac{\langle H(\omega, t)\rangle_{\Delta t}}{n(\omega)}]^2 dt$$

Degree of statistic stationary is described as follows

$$DSS(\omega, \Delta t) = \frac{1}{T}\int_0^T [1 - \frac{\langle \overline{H(\omega, t)}\rangle_{\Delta t}}{n(\omega)}]^2 dt$$

Properties

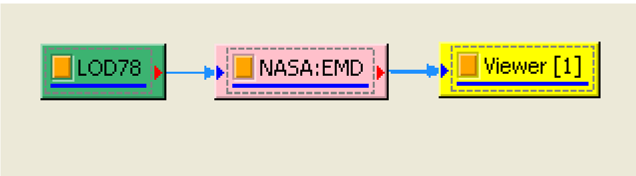
This module accepts input of Spectra (which could be real number, single channel, Regular). The output format is real, single channel, and Regular signal.

Option	Meaning	Default
TimeAverage	TimeAverage is the time span to average over of the spectra data. The unit of the time span is samples.	19

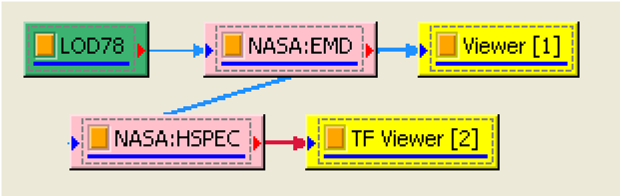
Example

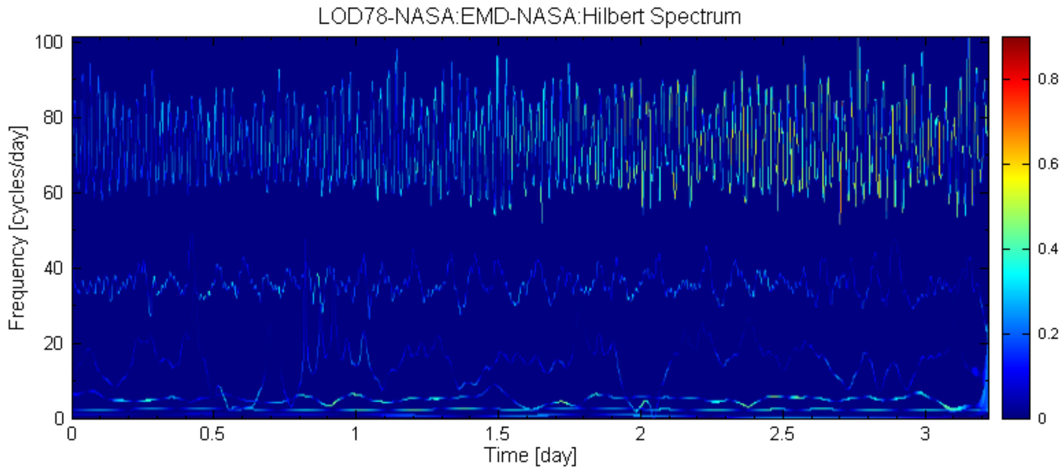
In this example, LOD78 is decomposed and then all IMFs are transformed into spectra by NASA Hilbert Spectrum. Lastly, use NASA DS to observe variation of frequencies.

1. Still use Source→Import data from file to read tfa file LOD78.tfa, in the installation directory (default to be C:\ProgramFiles\DynaDx\DataDemon\demo\HHT). Next perform Compute→NASA HHT→NASA EMD, and LOD78 would be decomposed into many IMFs.

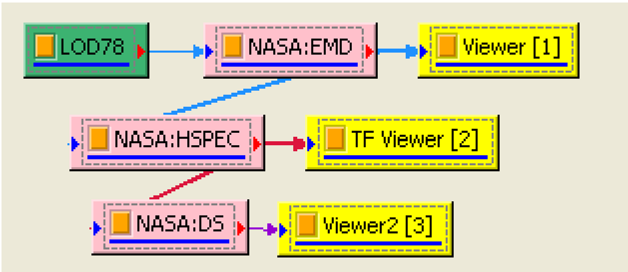


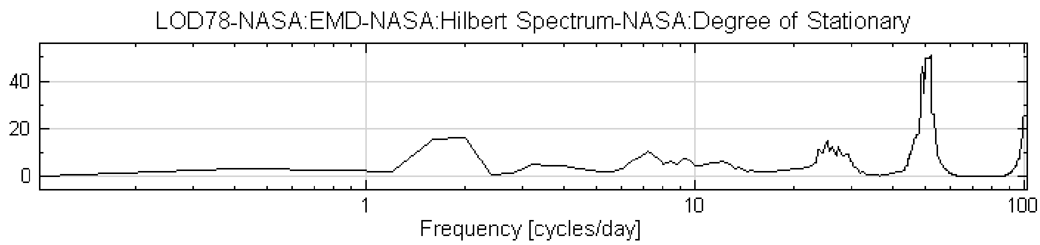
2. Then use Compute→NASA HHT→NASA Hilbert Spectrum to connect NASA EMD, and furthermore time-frequency result is displayed in the TFA Viewer.





3. Lastly, connect Compute→NASA HHT→NASA Degree of Stationary with NASA Hilbert Spectrum, and then display the result of NASA Degree of Stationary in the Channel Viewer. Set the Properties/XaxiaType as LogAxis.





References

1. N. E. Huang, Z. Shen, and S. R. Long, et al, “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary Time Series Analysis,” Proceeding of Royal Society A, vol. 454, pp. 903-995, 1998

2. Semion Kizhner, Thomas P. Flatley, Dr. Norden E. Huang, Karin Blank, Evette Conwell “On the Hilbert-Huang Transform Data Processing System Development”, 2004 IEEE Aerospace Conference Proceedings, Big Sky Montana, March 6-13, 2004

3. N. E. Huang, Z. Wu, S. R. Long, K. C. Arnold, K. Blank, and T. W. Liu, “On instantaneous frequency,” Advances in Adaptive Data Analysis Vol. 1, pp. 177-229, 2009

4. Wu, Z., and N. E. Huang, "Ensemble Empirical Mode Decomposition: a noise-assisted data analysis method," Advances in Adaptive Data Analysis, Vol. I, No. I, July 24, pp. .1-41, 2008

Glossary

Analytical signal - A complex valued signal the results from applying the Hilbert Transform. The real part is the original signal; the imaginary part is the complex conjugate.

Cauchy principal value - The value of the integral evaluated with the contribution from the singular point within a neighborhood of a vanishing radius excluded. The internet encyclopedia, Wikipedia, has a good article on the Cauchy principal value at:

http://en.wikipedia.org/wiki/Cauchy_principal_value

Cohen's paradoxes - First, instantaneous frequency may not be one of the frequencies in the spectrum... Second, if we have a line spectrum consisting of only a few sharp frequencies, then the instantaneous frequency may be continuous and range over an infinite number of values. Third, although the spectrum of the analytic signal is zeros for negative frequencies, that instantaneous frequency may be negative. Fourth, for a band limited signal the instantaneous frequency may go outside the band" (Cohen, 40).

Curvature - A calculation to determine how much an object derivate from being flat. Used in the curvature sift.

Degree of stationarity - How much a signal's statistics vary over time.

Empirical Mode Decomposition - An algorithm that decomposes data into Hilbert Transform-friendly signals, known as Intrinsic Mode Functions

Extrema - A local maximum or minimum in a signal. For example, in a signal sin(x), there would be a maximum extrema at pi/2 and a minimum extrema at (3*pi)/2

Filter - A device that creates a new signal based on the old one by permitting only certain properties of the original signal to come through. Traditional filters are non-adaptive, functioning by cutting off specific frequencies for the entire signal.

Fourier transform - The Fourier transform converts a signal into a sum of multiple sinusoidal functions. For information on the Fourier transform, please refer to: http://en.wikipedia.org/wiki/Fourier_transform

Generalized zero crossing - A method of calculating local frequency values using the distances between zero crossings and extrema.

Gibbs phenomenon - Often seen in results after a Fourier transform is applied, there is a series of oscillations that permeate data in areas of discontinuities. As the Hilbert transform is implemented through two Fourier transform proposed by Gabor, Gibbs phenomenon will also show up whenever the end of the data when spliced showed a jump.

Hilbert, David - Renowned mathematician for who first observed the functions later named Hilbert transform by Hardy. For more information, refer to:

http://en.wikipedia.org/wiki/Hilbert

Hilbert Transform - Used to derive the analytical signal from data. This information can be used to calculate the instantaneous frequency.

Hilbert-Huang Transform - A system for obtaining the instantaneous frequency of a signal. It involves applying the Empirical Mode Decomposition algorithm to the data to derive the Intrinsic Mode Functions, which the Hilbert Transform is then applied to calculate the instantaneous frequency.

Huang, Norden E. - Norden E. Huang is a senior fellow at NASA, Goddard Space Flight Center. He holds a doctoral degree in Fluid Mechanics and Mathematics from the Johns Hopkins University. Dr. Norden Huang has worked on nonlinear random ocean waves spectrum determination. Recently, he developed a new method, the Hilbert Spectrum Analysis, specifically to process non-stationary and nonlinear time series. He developed the Empirical Mode Decomposition method that is the basis for the Hilbert-Huang Data Transform. Dr. Norden Huang is a member of the National Academy of Engineering. (Kizhner, 2004)

Instantaneous frequency - Frequency is often defined as the reciprocal of the time length of an event (such as the period). Instantaneous frequency is the ability to calculate the frequency based on instantaneous rate of change of the phase function. For a simple oscillation, the instantaneous frequency is identical to the traditional frequency defined as the reciprocal of the period. The instantaneous frequency provides sharper, more local results for describing nonstationary and nonlinear processes.

Intermittency - A natural phenomenon in turbulence when certain scale of motion will occur sporadically. Here we use it to designate the method of removing a sporadic signal from a given IMF.

Intrinsic Mode Function - Any function that satisfies the following conditions: that the number of extrema differ from the number of zero crossings by no more than one, and that the mean of the envelopes defined by the maxima and minima is zero. IMFs

are generated using the Empirical Mode Decomposition process, and are useful for obtaining the instantaneous frequency via the Hilbert Transform.

Linear - The conditions as given in the linear system.

Marginal Spectrum - The amount of energy in the signal for each frequency.

Noise - Data that is either without meaning or relevance to the information of interest.

Normalized Hilbert Transform - A method of normalizing the amplitude of the data before applying the Hilbert Transform in order to satisfy the condition stipulated by the Bedrosian theorem.

Orthogonality - Defined as the inner product of the two vector is identical zero. Here we use it to check how linearly dependent the generated IMFs are. Values closer to zero are "better".

Residual - A signal that is the result of a series of IMFs being subtracted from the original signal. A residual which is created when all possible IMFs being subtracted from it has the property of having no more than 2 extrema.

Riding Wave - Any occasion when there are multiple extrema in between to consecutive zero-crossings..

Sifting Criteria - The number of times a signal needs to pass the test for an IMF in the EMD sift process before it can be considered an IMF.

Spectrum - A two dimensional plot of frequency verses time, with color intensity used to illustrate the amount of energy at each point of the signal.

Stationary - Demonstrating a lack of change.

Stopping Criteria - The rules used to determine when to stop sifting when generating an IMF. This includes a test to see if the signal meets the definition of an IMF, as well as the sifting criteria.

Threshold - The numerical value used during intermittency to determine if a section of signal between two zero-crossings should be kept or discarded. Distances between zero-crossings less than the threshold are kept; otherwise, they are zeroed out. Threshold is measured in number of samples, without regard to the timescale of the signal.

Zero Crossing - Where the signal crosses the x-axis at y = 0.

3.7.4 RCADA Instant. Frequency

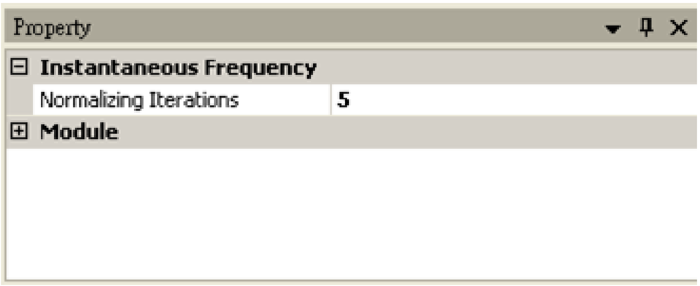
RCADA Instant. Frequency is a method for calculating instaneous frequency provided by RCADA. This module uses the same algorithm as Instant. Frequency function in MATLAB code from RCADA.

Introduction

Please refer to http://rcada.ncu.edu.tw/research1.htm

Properties

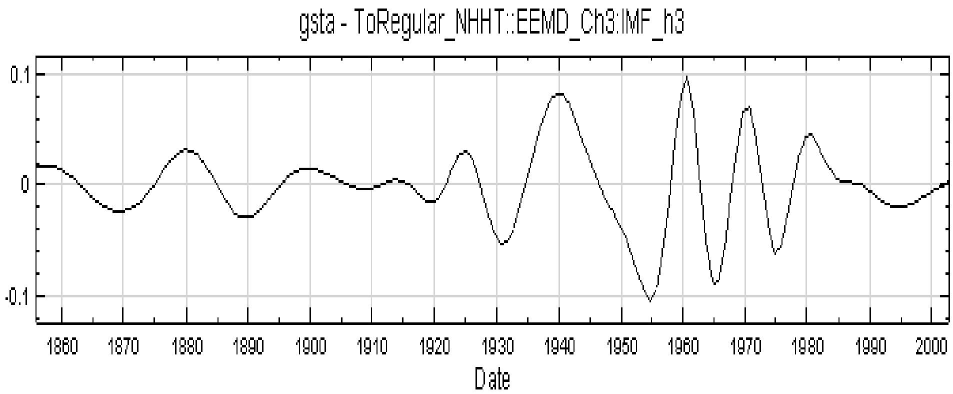
This module accepts input signals of real number, single channel or multi-channel, regular, and audio.

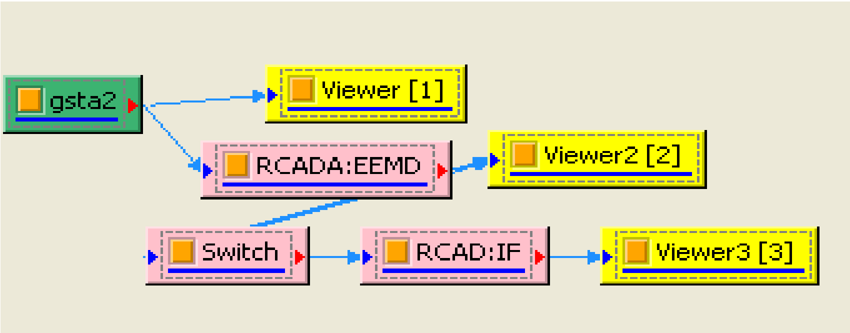


Property Name	Property Definition	Default Value
Normalizing Iterations	Number of times to obtain envelop maxtrima	5

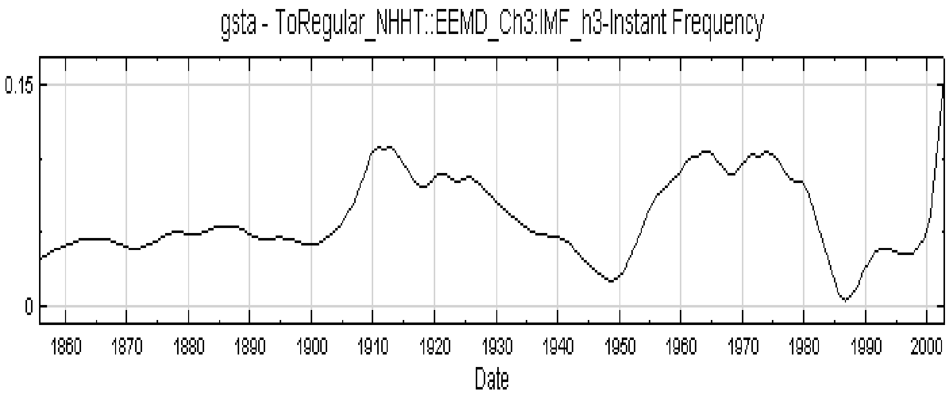
Example

Continue the analysis for the gsta.dat after RCADA EEMD decomposition. Using Compute / Channel / Channel Switch to get the 3rd IMF, and connect it to HHT / RCADA Instant. Frequency, display the result with Channel Viewer. Please refer to demo68 in C: \ Program Files \ AnCAD \ Visual Signal \ demo \ HHT.





The instaneous frequency is:



Related Function

RCADA Instant frequency, RCADA Spectrum, Hilbert Transform.

References

http://rcada.ncu.edu.tw/research1_clip_reference.htm

Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung and Henry H. Liu: “The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-Stationary Time Series Analysis”, Proceedings of the Royal Society, Vol. 454, No. 1971, 1998.

Huang, N. E., M. L. Wu, S. R. Long, S. S. Shen, W. D. Qu, P. Gloersen, and K. L. Fan (2003): “A confidence limit for the Empirical Mode Decomposition and Hilbert Spectral Analysis”, Proc. Roy. Soc. London, 459A, 2317-2345.

3.7.5 RCADA Spectrum

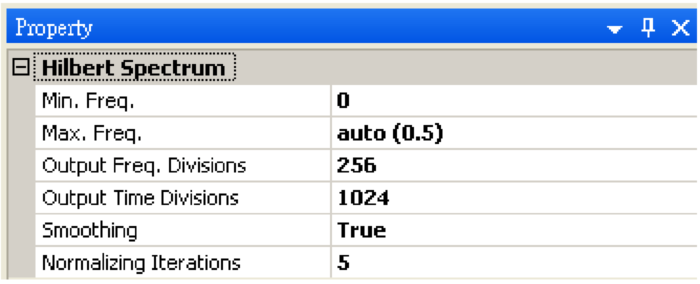
RCADA Spectrum is Hilbert Spectrum method provided by RCADA.

Introduction

Please refer to http://rcada.ncu.edu.tw/research1.htm, or 3.5.4 Hilbert Spectrum section.

Properties

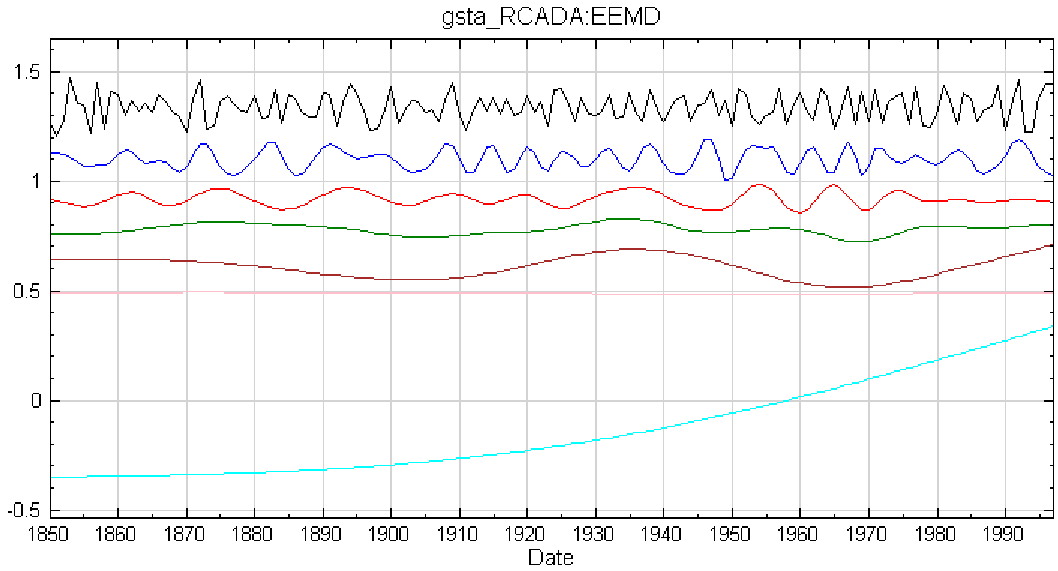
This module accepts input signal of real number, single channel or multi-channel, regular, and audio.

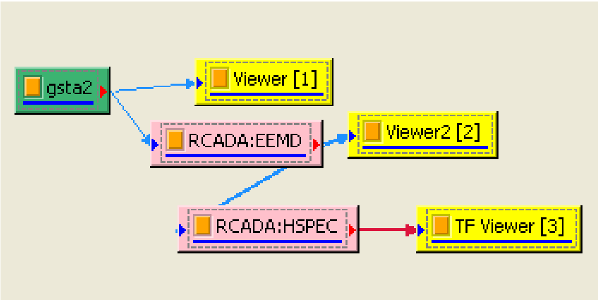


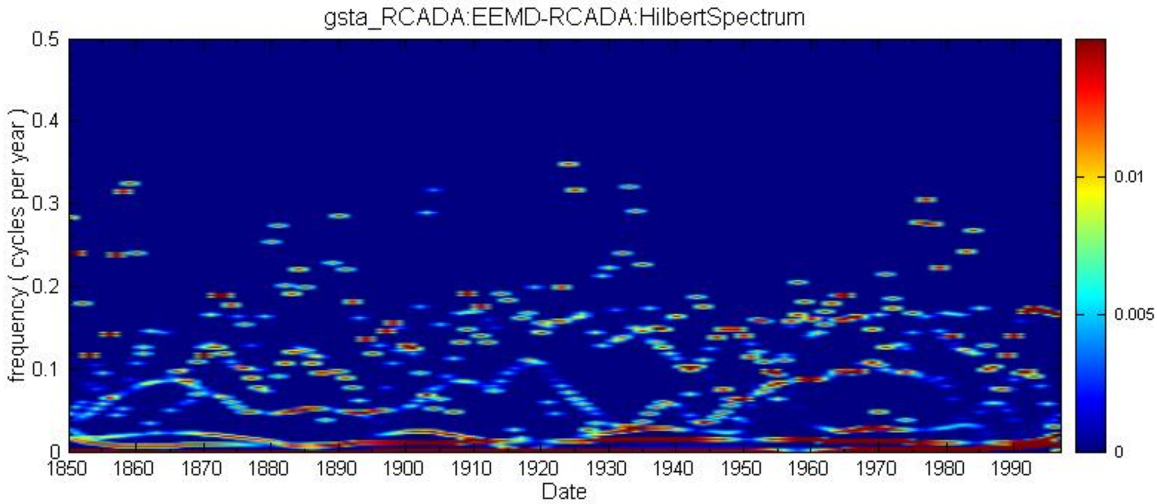
Property Name	Property Definition	Default Value
Min. Freq	Start value for Instant. Frequency	0
Max. Freq	End value for Instant. Frequency	0.5 * sampling frequency
Output Freq. Divisions	Number of grid in frequency axis for the spectra plot	256
Output Time Divisions	Number of grid in time axis for the spectra plot	1024
Smoothing	Smoothing the curve with Gaussian function	True
Normalizing Iterations	Number of times to obtain envelop maxtrima	5

Example

Continue the analysis for the gsta.dat after RCADA EEMD decomposition. Connect the results to Compute / HHT / RCADA Spectrum and display them with Viewer / Time Frequency Viewer. Please refer to demo68 in C: \ Program Files \ DynaDx \ DataDemon \ demo \ HHT.







Related Function

RCADA EEMD, Hilbert Spectrum.

References

http://rcada.ncu.edu.tw/research1_clip_reference.htm

Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung and Henry H. Liu: “The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-Stationary Time Series Analysis”, Proceedings of the Royal Society, Vol. 454, No. 1971, 1998.

Huang, N. E., M. L. Wu, S. R. Long, S. S. Shen, W. D. Qu, P. Gloersen, and K. L. Fan (2003): “A confidence limit for the Empirical Mode Decomposition and Hilbert Spectral Analysis”, Proc. Roy. Soc. London, 459A, 2317-2345.

3.8 Enhanced (Professional Only)

Fast Short-Term Fourier Transform： STFT’s fast version with less memory requirement.

Remove Bump：Remove bumps or jumps from the signal and connect them with smooth line for recovering the original signal.

Fast Trend Estimater：Trend Estimater’s fast version.

Fast Iterative Gaussian Filter：Iterative Gaussian Filter’s fast version.

Fast MSE：MultiScale Entropy’s fast version.

Peak Detection： detect the peak position of the signal or compute the time difference between the peaks.

R-R interval：Compute time difference of two R waves of the ECG signal.

Teager：Compute the instantaneous frequency and amplitude of the signal.

Rolling MSE：Viewing the variation of the amplitude by different time and scales.

PCA：decompose the composite signal to single signals.

ICA：decompose a composite signal to a statistical independent signal.

3.8.1 Fast Short-Term Fourier Transform

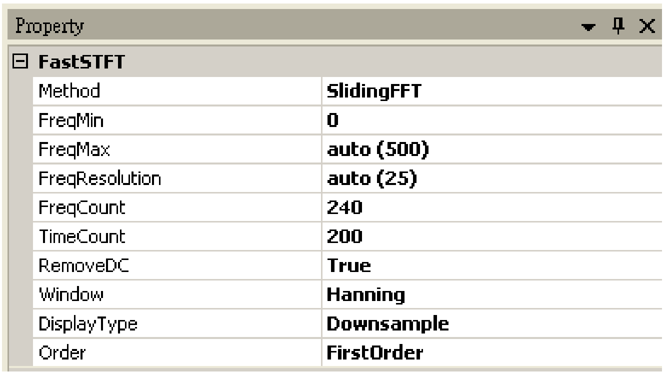
The Fast Short-Term Fourier Transform has the same functions with Short-Term Fourier Transform, but is faster than the original version.

Introduction

This module is the fast version of the STFT: in terms of the same size data, the module consumes less time and memories. It shows hundreds times speedup; in terms of 3Gb memory (the upper limit of the Windows XP 32bit), the computation upper limit of the standard STFT is about 4 million data points; as for the fast STFT, the limit is about 22 million data points.

Properties

This module accepts real numbers, single channels, regular signals and audio inputs; the format of the input signal is the plural and single channel spectra data. Properties are set up as the below table:

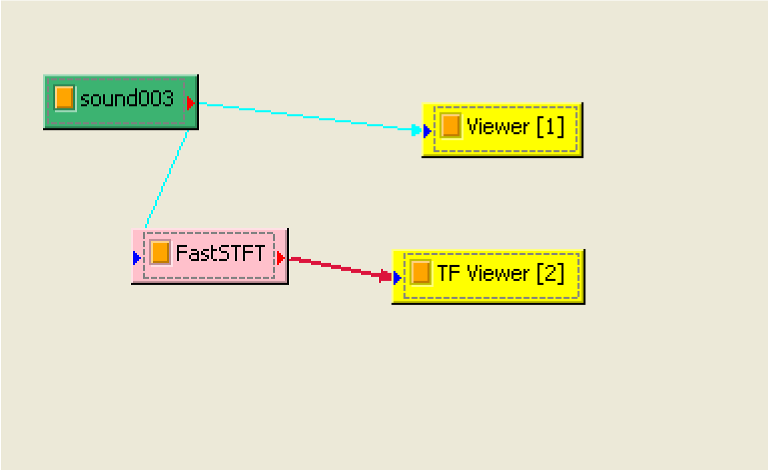


Property Name	Property Definition	Default Value
Method	Setting the method of computations, including FFT, SlidingFFT. FFT provides the 1st Order Solution, SlidingFFT provides the 1st, 2nd Order Solution. SlidingFFT moves one point for each computation, so the computation grid is at high densities but not be presented fully on the screen. If the FreqCount or the TimeCount is set less than the number of the time/frequency grid of the data, the users need to set the DisplayType.	SlidingFFT
DisplayType	In the results of the SlidingFFT, if the FreqCount or the TimeCount is less than the computation grid of	Downsample

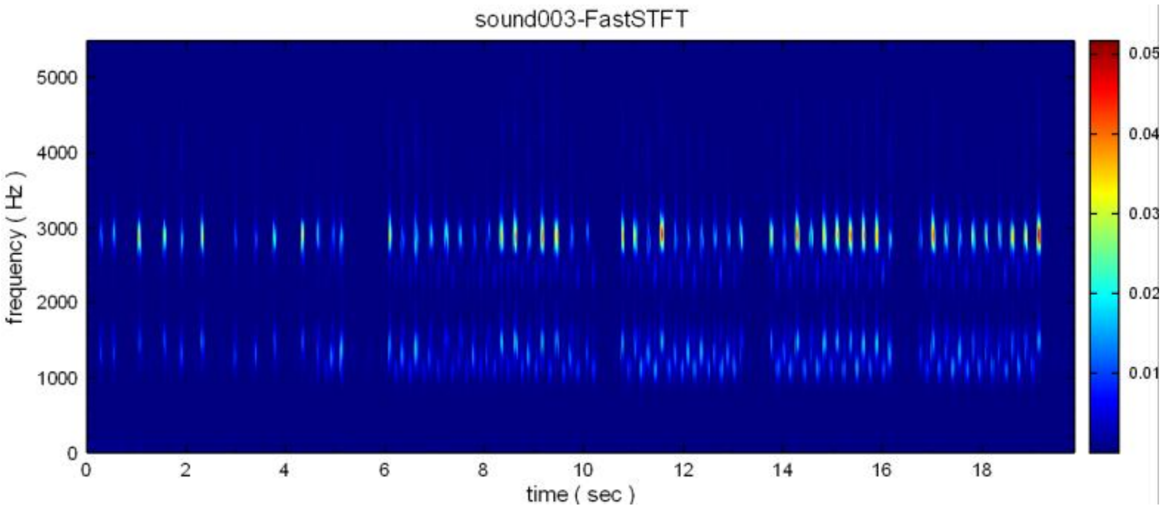
	the data, the sampling points are reduced by the Downsample, the Maxima, the Minima and the Average in the range of the data's time or frequency.	
Order	When the method is the SlidingFFT, the users could set the First or the Second Order Solution. As for the FFT, the choice is just the First Order.	FirstOrder
FreqAxis	The distribution of the circular frequency could be the linearAxis or the LogAxis. The LogAxis is usually used audio analysises.	LinearAxis
FreqMin; FreqMax	By these properties, the users could set the up and down borders of frequency in the figure.	0; 0.5*(Sample Frequency)
FreqResolution	This value affects the size of the window function. The lower the setting is, the longer the window function is.	(Sample Frequency) /40
FreqCount	Showing the grid number of the STFT in the direction of the frequency.	Auto
TimeCount	Showing the grid number of the STFT in the direction of the time.	Auto
RemoveDC	This value decides whether the DC signal is removed before the STFT analysis.	True
Window	During the time-axis segmentation, the STFT filters the signal by the window function.	Gaussian

Example

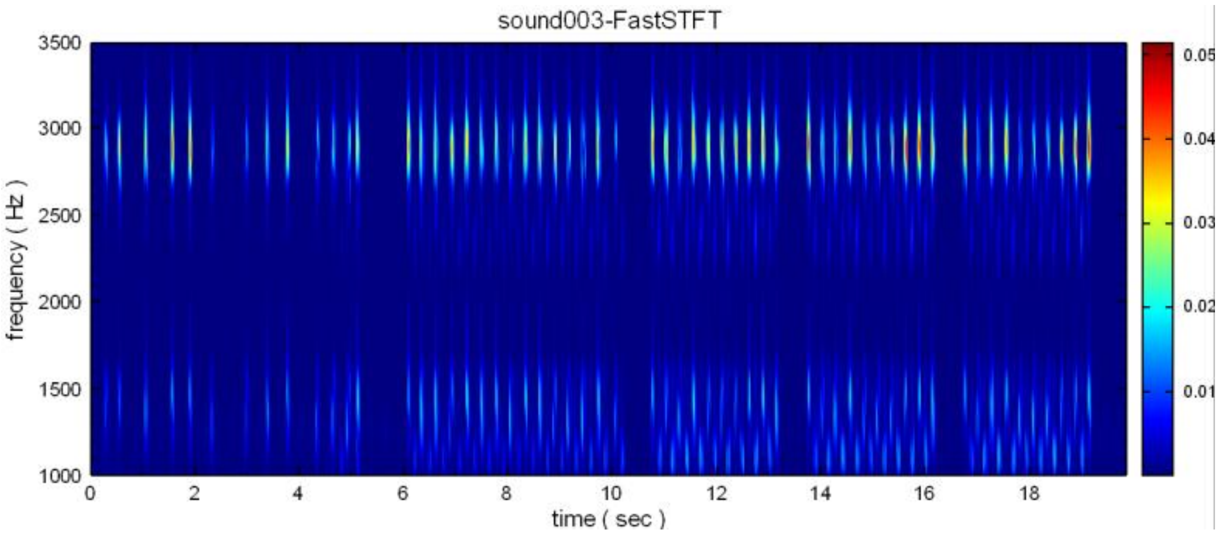
The configuration of the mudule is as below:



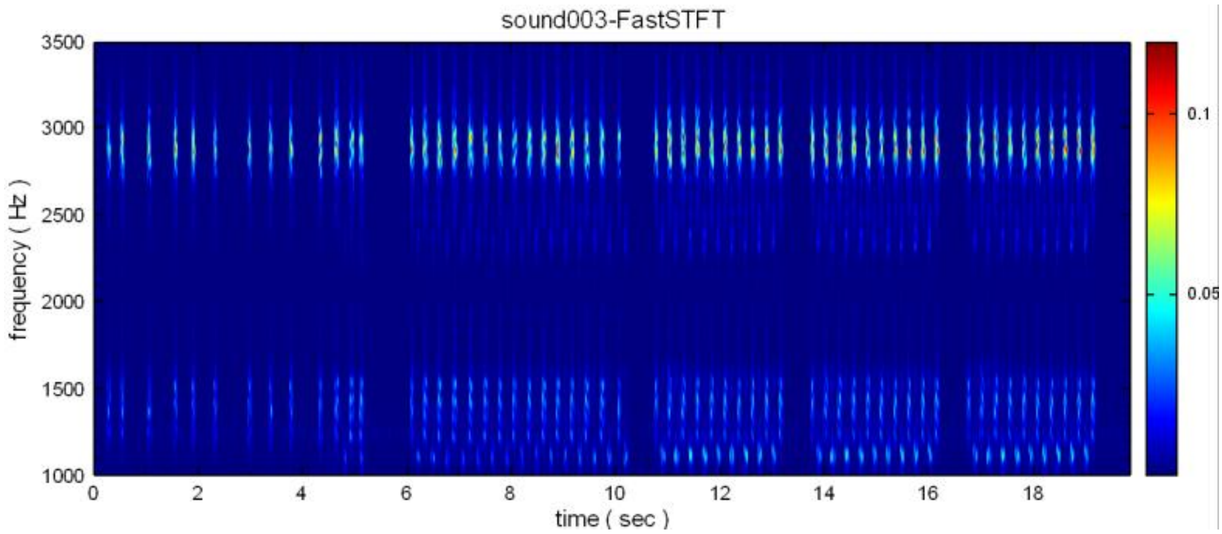
The result of the analysis is as below:



By modifying the value of the FreqMax and the FreqMin, the users can get the results in the range of 1000Hz to 3500Hz.



By modifying the Frequency Resolution to 75 (the original value is 225), the result is as below:

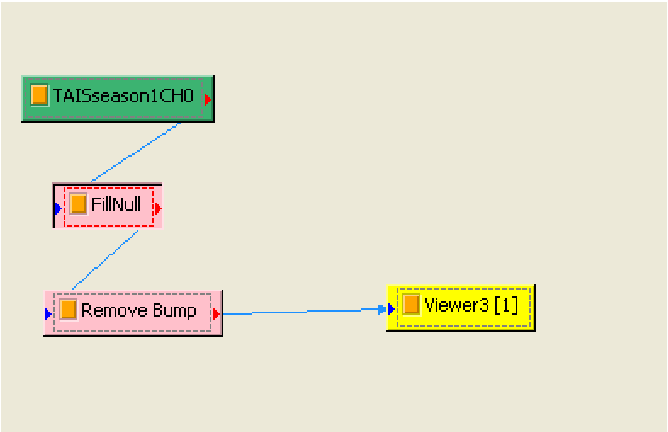


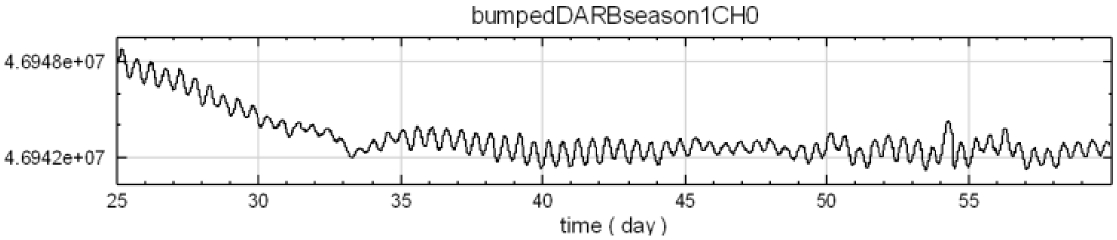
We can find that the frequency resolution is better.

the seismometer data

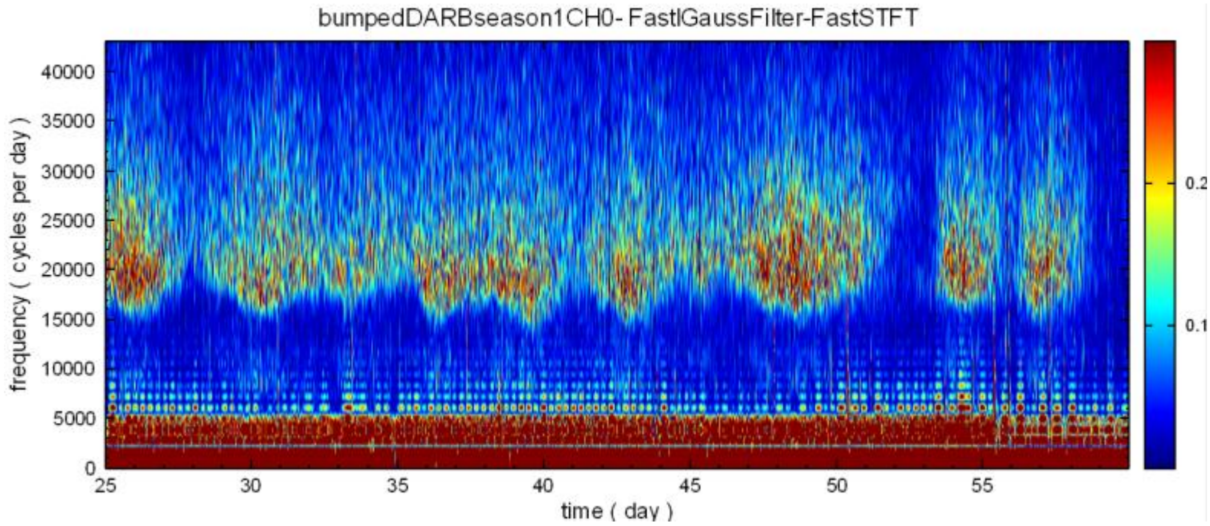
The seismometer signal has 8 million points. If the signal is processed by the standard STFT, the 3G memory is not sufficient.

After the interpolation and the RemoveBump process, the result is as below:

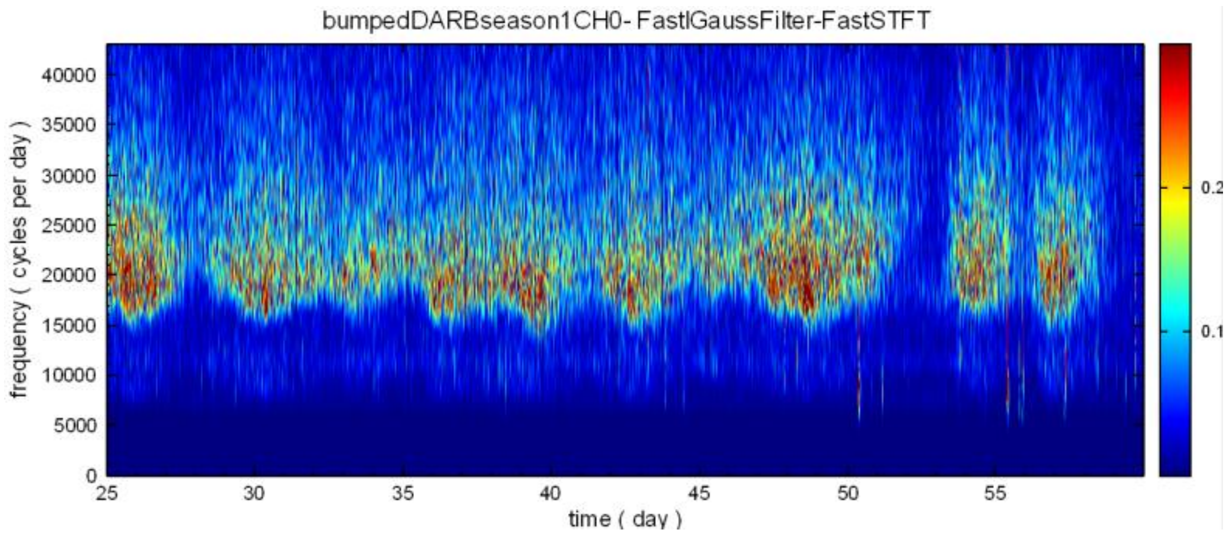




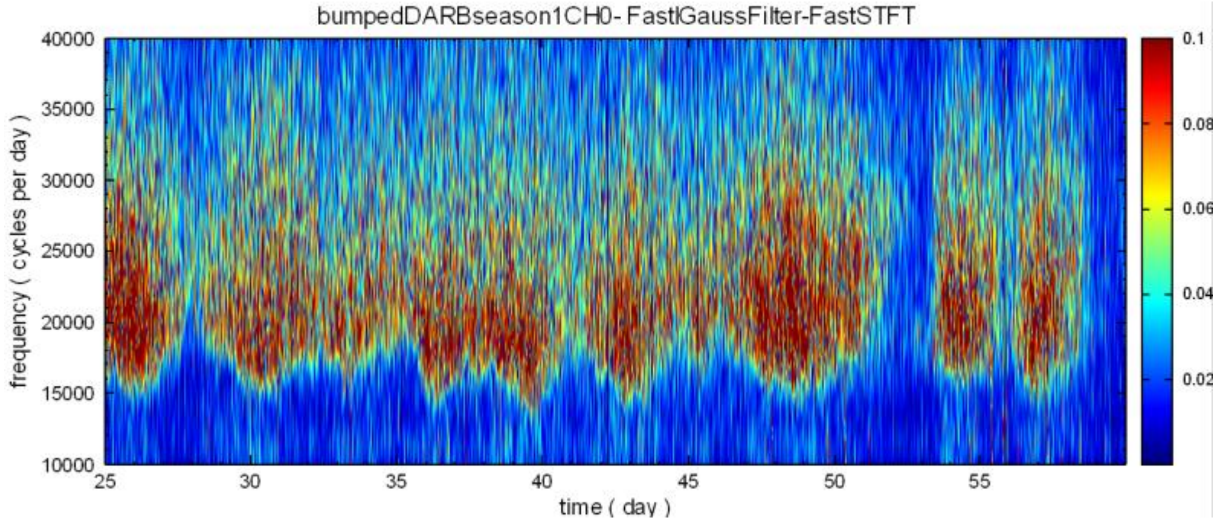
After linking this signal to the Fast Trend and then the Fast STFT (using the default setting), the result is shown as below:



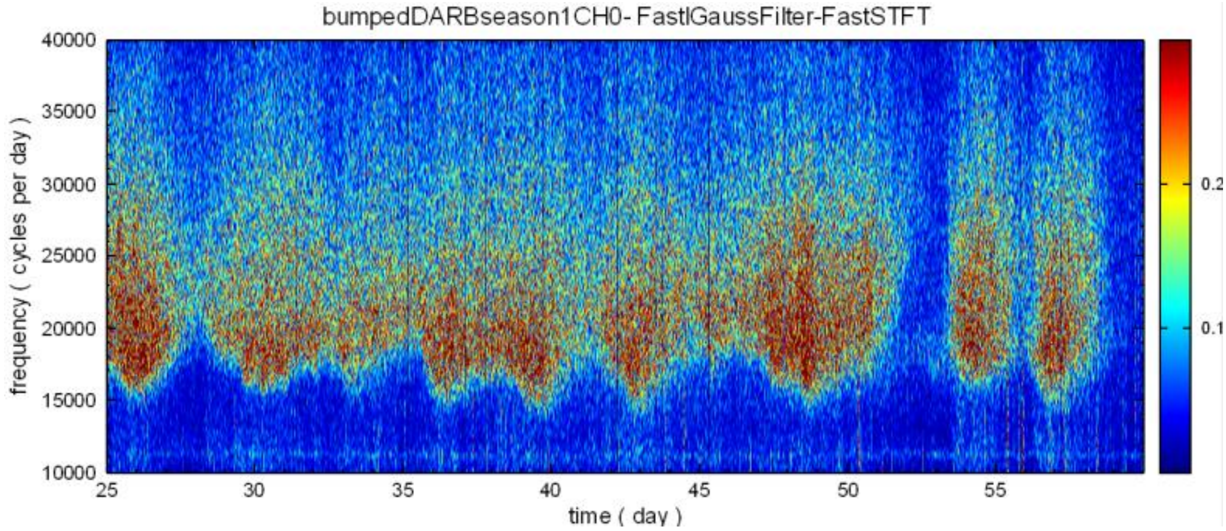
We can find that the low frequency is not filtered off completely, so the trend frequency of the Fast Trend is tuned to 3000. The result is shown as below:



We can get the signal in the range of 10000Hz to 40000Hz by resetting the FreqMax and FreMin. The result is shown as below:



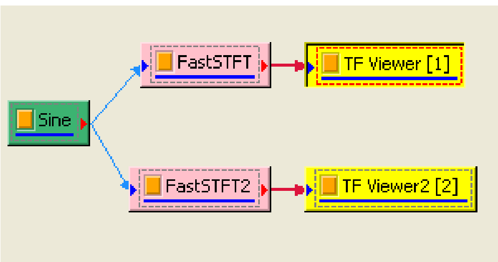
We can find that the frequency resolution is not good. By tuning the Frequency resolution to 500 (the original value is 2500), the result is modified as below:

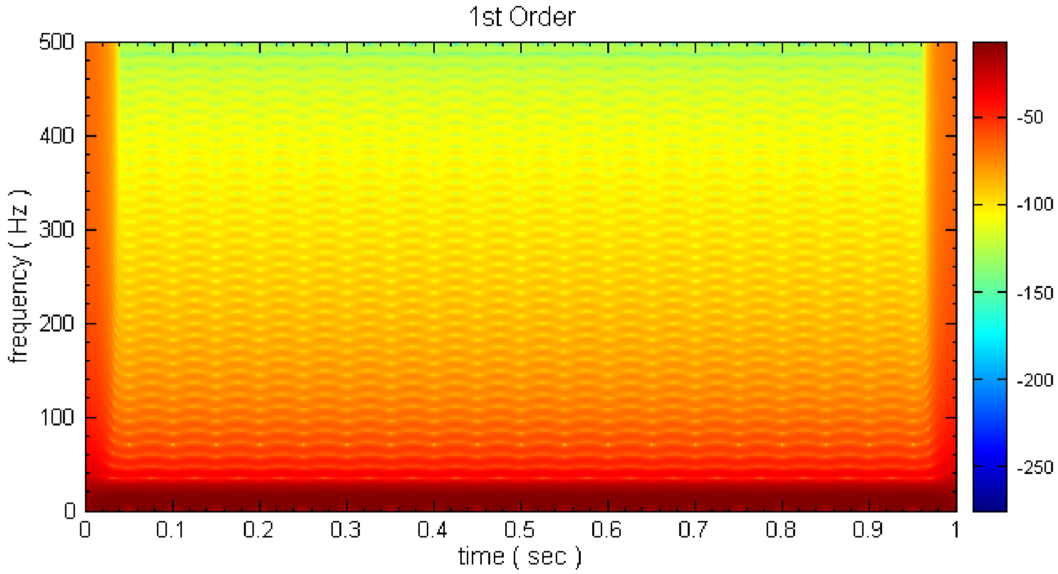


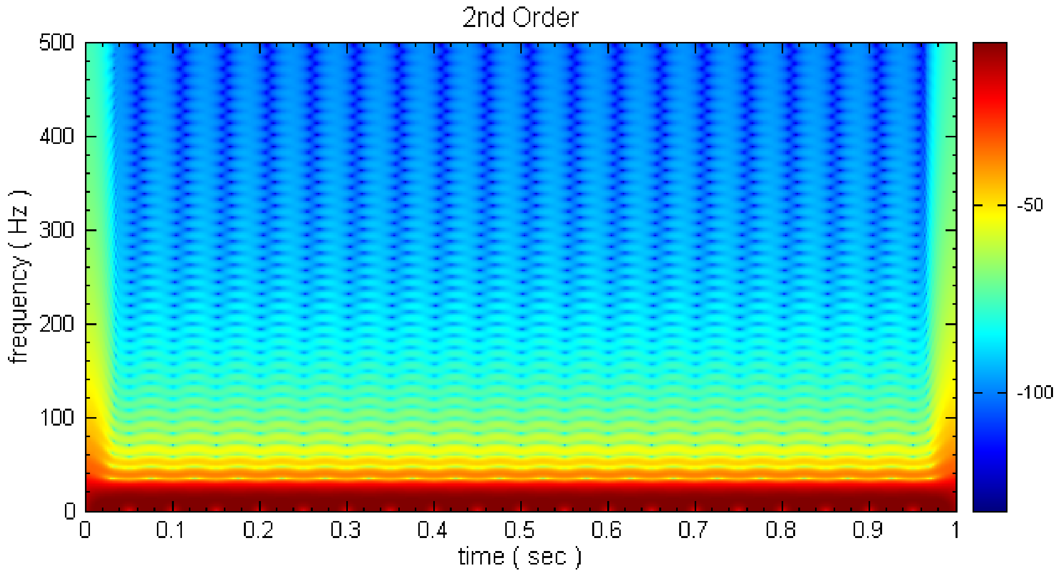
Comparing the results of the First Order and the Second Order

Generating the Sine signal by the Source / Sine Wave, and then linking it to two FastSTFT, the result is shown by two TF Viewers. The YValueType of the TF Viewer is set to Gain.

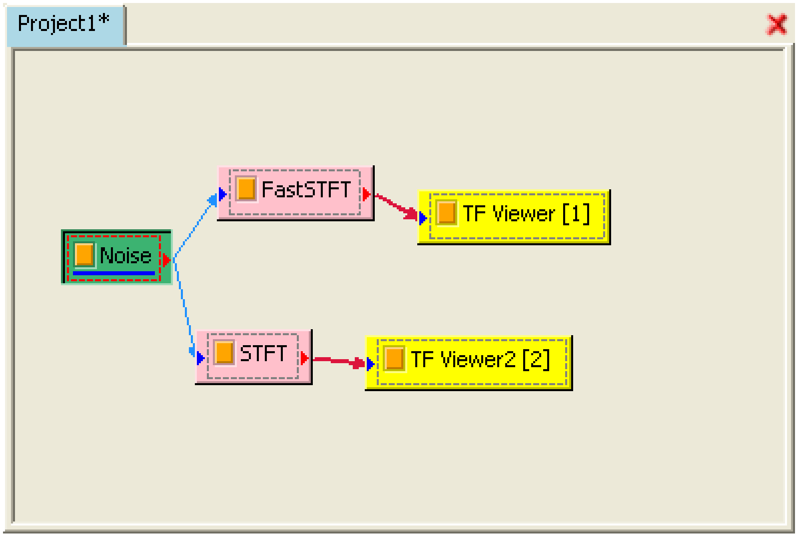
From the TF Viewer, we can find that the computed amplitude of the Second Order is less the First Order in the high frequency.







Analysing the data including Noise by the Fast STFT and the standard STFT.



The same data is analyzed by the FastSTFT and the STFT. Based on the 2.8GHz (Intel E6300 Dual Core) computer, the computation times are 0.35 second and 21.84 second.

Setting the Frequency Resolution to 250, the computation time of the FastSTFT is 0.3593 second, the computation time of the standard STFT is 64.2343 second.

Related Functions

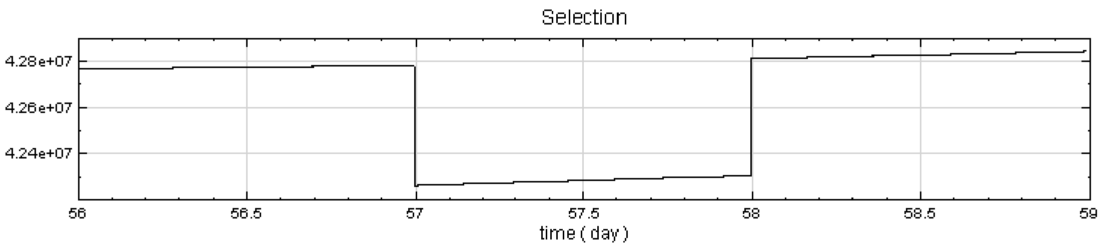
Short term Fourier Transform, Enhanced Morlet Transform.

Reference

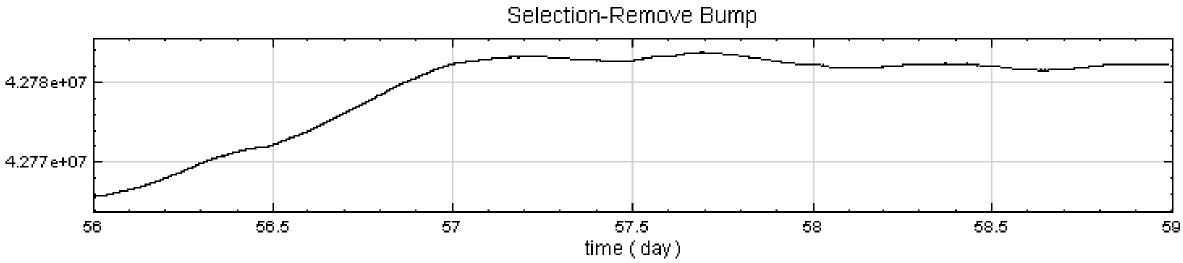
A Wavelet Tour of Signal Processing (2nd Ed).

3.8.2 Remove Bump

Due to the measuring hardware, such as calibarion and baseline deivation, the collected signals could have discontinuities as Bumps or Jumps shown below.



This module removes these Bumps and then smoothly reconstructs the signal:



Introduction

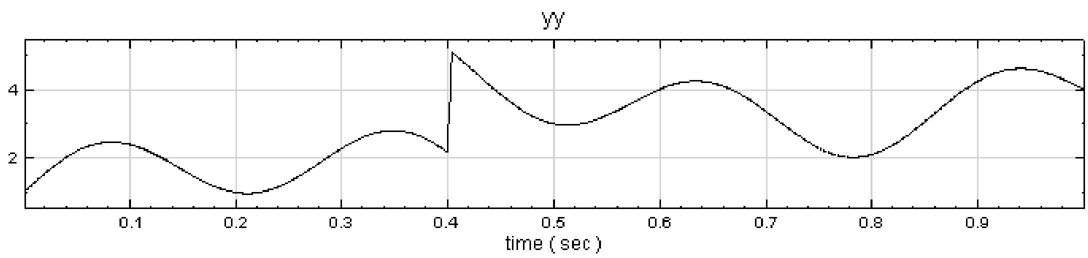
This module decides whether Bumps or Jumps happen by the intersection of two conditions:

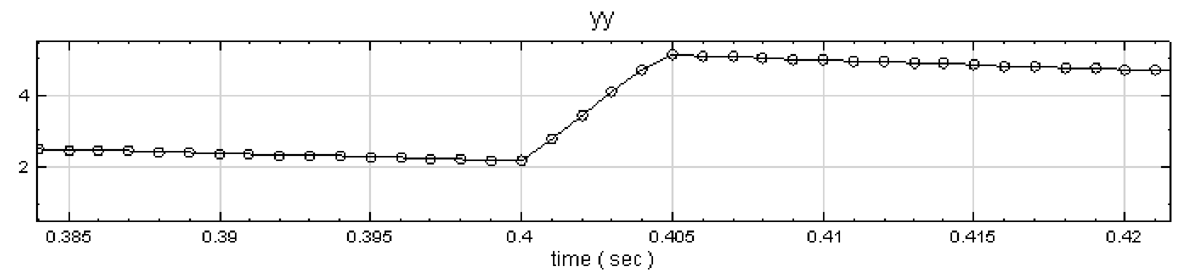
If there are a series of values that exceed the critical value S_{crit} and increase/decrease continuously and progressively.

If the gradient of the data point exceeds the critical gradient J_{crit}.

Here we show the difference of the J and the S:

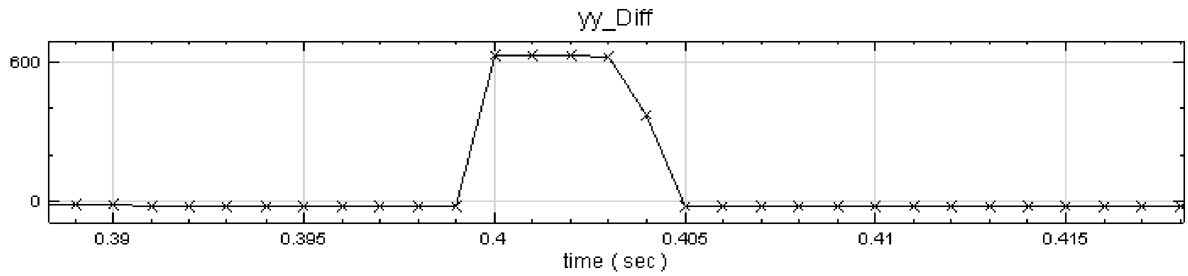
There is a Bump in the signal at the time of 0.4 second. Zooming in this zone, we can find that they are five discrete points.





In these six points (t =0.4s 到 t = 0.405s， y 從 2.1756 到 5.0575), the maximum is 5.50575 - 2.1756 = 3.33015。

In these six points, the S value is shown as below:



(The Compute / Math / Diff gives the gradient.)

If the S and J values are both exceed the critical value that is the Bump which is needed to be corrected.

Properties

This module accepts real numbers, single channels, regular signals and audio inputs; the format of the input signal is the plural and single channel spectra data. Properties are set up as the below table:

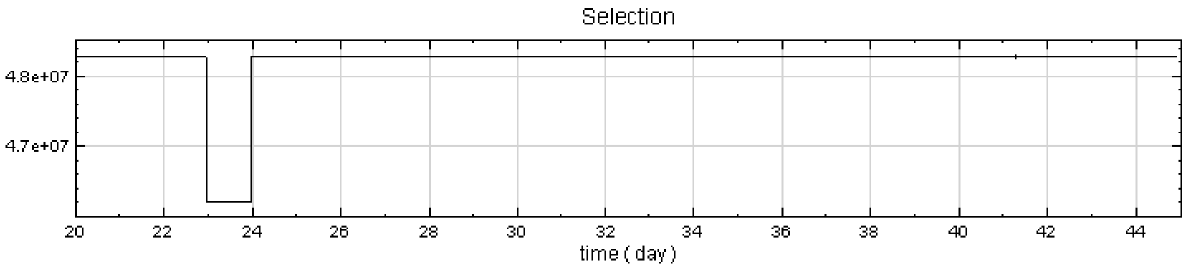
Property	
⊞ Module	
⊟ Remove Bumps	
Output Type	Signal
Jump Threshold	0.5932954172138134
Jump Threshold Ratio	0.3
Slope Threshold	593.29541721381338
Slope Threshold Ratio	0.3
Number of iterations	1
StartPosition	0
EndPosition	end (0.132)

Property Name	Property Definition	Default Value
Output Type	Signal: removing the signal after the Bump or the Jump. Bump: the Bump or the Jump signal	Singal

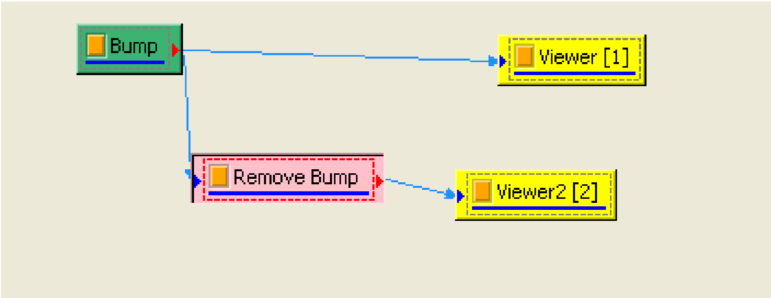
Jump Threshold	If the increasing/decreasing value exceeds this threshold, this point is considered as the candidate point.	90% of the maximum value of the entire signal
Jump Threshold Ratio	If the maximal increasing/decreasing value is J_{max}, $J_{crit}=J_{max}\times J_{ratio}$	0.3
SlopeThreshold	If the absolute value of the gradient exceeds this threshold, this point is considered as the candidate point.	90% of the maximum gradient of the entire signal
Slope Threshold Ratio	If the maximal absolute value of the gradient is S_{max}, $S_{crit}=S_{max}\times S_{ratio}$	0.3
Number of iterations	If properties are not appropriate, the module may not remove the Bump or Jump completely. This number controls the repeat time of searching the Bump or the Jump.	1
StartPosition	The start time of processing the Bump or the Jump	The start time of the entire signal
EndPosition	The end time of processing the Bump or the Jump	The end time of the entire signal

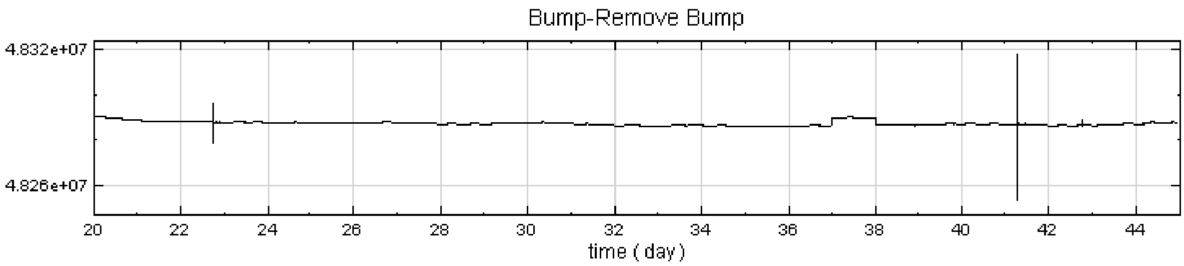
Example

The seismometer data

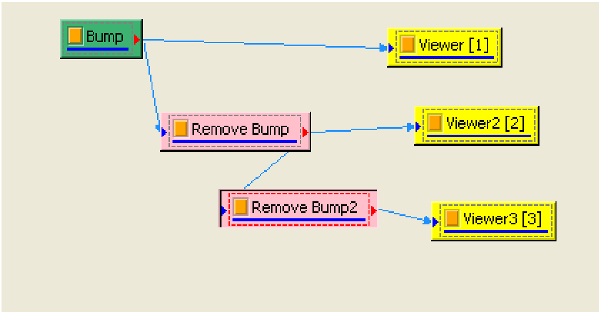


Linking the original signal to the Compute / Enhanced / RemoveBump, and then viewing the result by the Channel Viewer. All of properties are set with the default values as below:





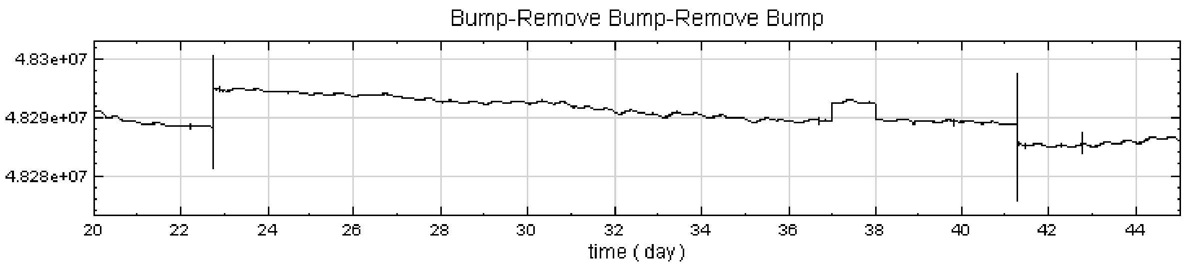
Where the two biggest Bump points are removed. After the first RemoveBump, we add another RemoveBump. The properties are set as below:



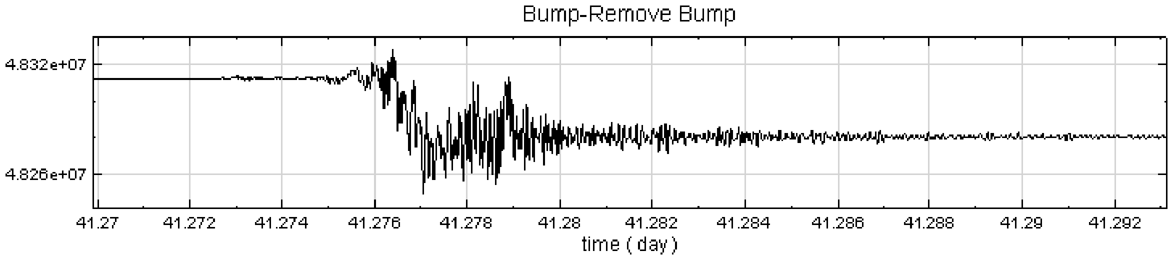
Setting the Jump Threshold Ratio and the Slop Threshold Ratio of the second RemoveBump to be 0.1 and other properties are set with the default values.

There are two processing methods:

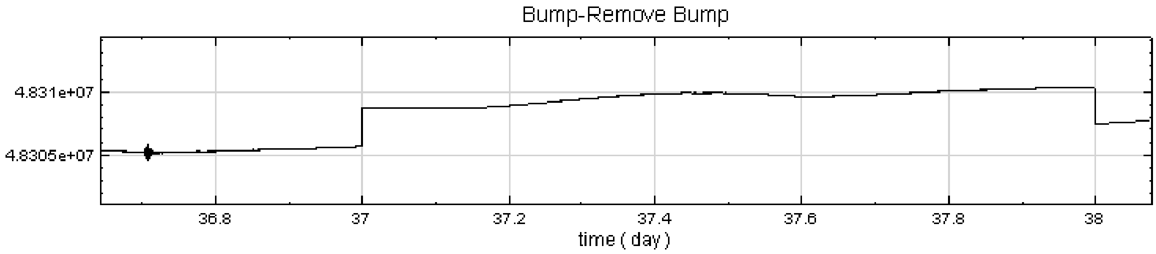
The first method:



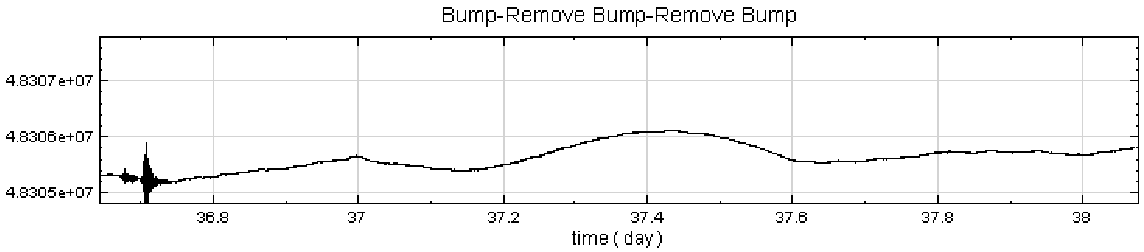
From the above figure we can see that there is a Bump between t = 22.77 and t = 41.27, but when we zoom in this area, it is not a Bump.



The Bump is in the time range of t = 37 to 38.

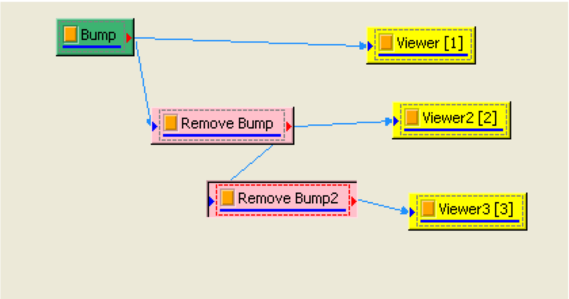


We can find that the Bump is removed in this area.

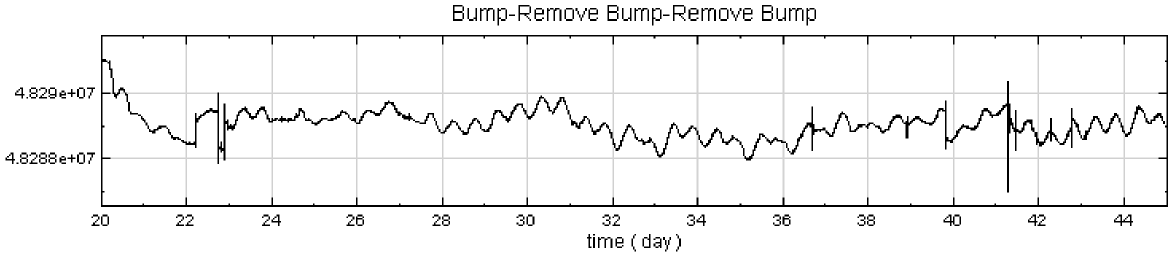


The second method:

Setting the Jump Threshold Ratio and the Slop Threshold Ratio to be 0.1 and remaining the StartPosition and the EndPosition unchanged:



The output is:



Related instructions

Data Selection, Diff.

3.8.3 Fast Trend Estimater

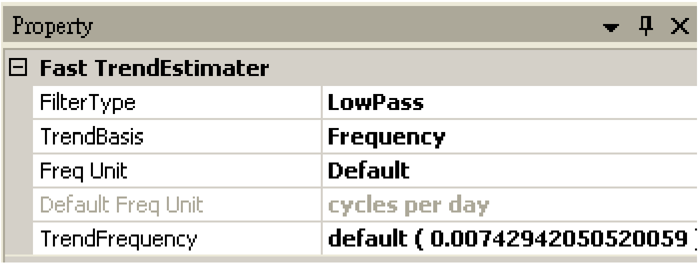
The Fast Trend Estimater is the same as the standard Trend Estimater, but the fast version and consuming less memory.

Introduction

Refer to the algorithms of the chapter: Iterative Gaussian Filter.

Properties

This module accepts real numbers, single channels, regular signals and audio inputs; the format of the input signal is the plural and single channel spectra data. Properties are set up as the below table:



Property Name	Property Definition	Default Value
Filter Type	It is the same as the Iterative GaussianFilter. LowPass：extracting the low frequency part of the signal. HighPass：extracting the high frequency part of the signal. ByPass：reserving all frequency parts of the signal.	LowPass
Trend Basis	Setting the properties based on the Period or the Frequency.	Frequency

If the Trend Basis is set to the Period, its properties are defined as below:

Property Name	Property Definition	Default Value
Trend Period	If the period exceeds the Trend Period, it is deemed as a trend signal. Corresponding to the Iterative Gaussian Filter:	0

	FL (= 2 /TrendPeriod) FH (= 4 / TrendPeriod)	
Time Unit	Setting the unit of the TrendPeriod.	Default
Default Time Unit	The default time unit of the input signal.	Sec

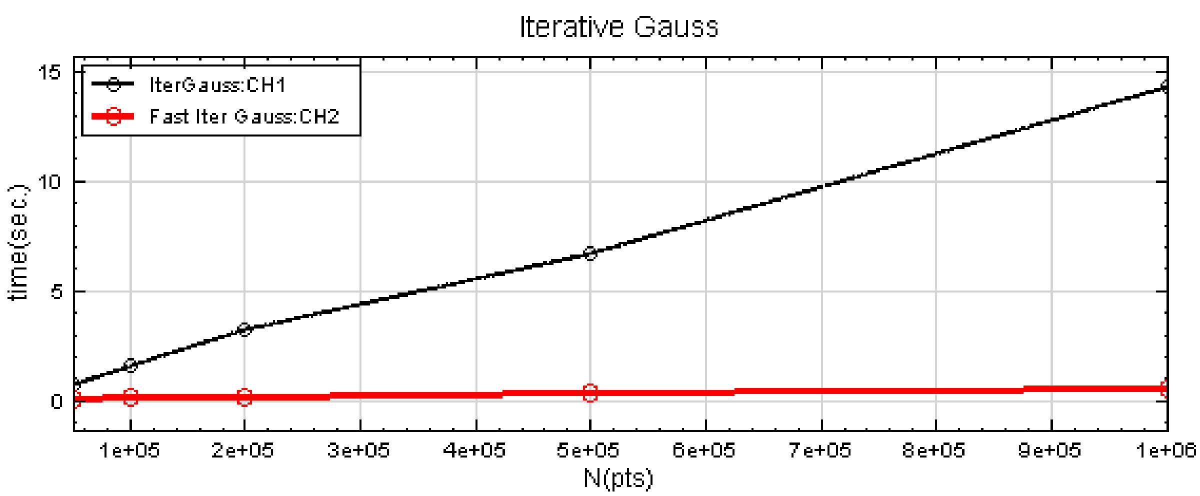
If the Trend Basis is set to the Frequency, its properties are defined as below:

Property Name	**Property Definition**	**Default Value**
Trend Frequency	If the frequency exceeds the Trend Frequency, it is deemed as a trend signal. Corresponding to the Iterative Gaussian Filter: FL (= 2 * TrendFrequency) FH (= 4* TrendFrequency)	0
Frequency Unit	Setting the unit of the TrendFrequency.	Default
Default Frequency Unit	The default frequency unit of the input signal.	Hz

Example

Please refer to the Trend Estimator.

Based on the computer with the CPU 2.8 GHz (Intel E6300), points by computing time figure for computing signals with Brown noises of different lengths by the standard/fast version Trend Estimater is shown as below:



Related instructions

Fast Iterative Gaussian Filter, Trend Estimater, Iterative Gaussian Filter.

References

Yih Nen Jeng, "Diffusive and Fast Filter Using Iterative Gaussian Smoothing", Department of Aeronautics and Astronautics, National Cheng Kung University

YIH-Nen Jeng P. G. Huang You-Chi Cheng, 2007, "Decomposition of one-dimensional waveform using iterative Gaussian diffusive filtering methods", Proc. R. Soc. A doi:10.1098/rspa

Yih-Nen Jeng, You-Chi Cheng, 2006, "Accuracy Comparison between Two Sharp and Diffusive Filters", Proc. R. Soc. A doi:10.1098/rspa

3.8.4 Fast Iterative Gaussian Filter

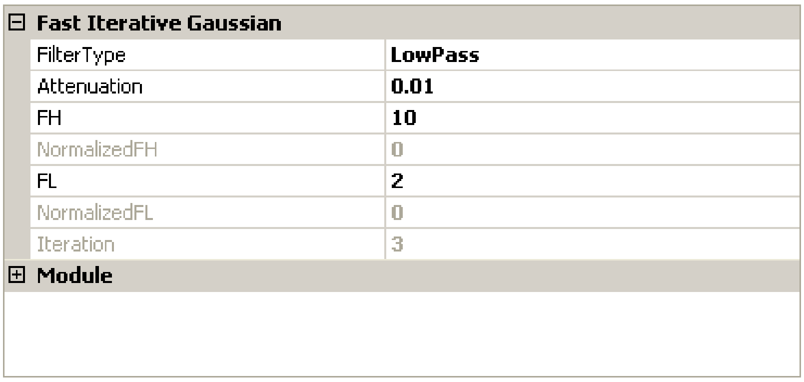
The algorithm of the Fast Iterative Gaaussian Filter is faster than the standard version, consuming less memory and presenting more accurate results especially on the boundary.

Introduction

Please refer to the introduction of the Iterative Gaussian Filter.

Properties

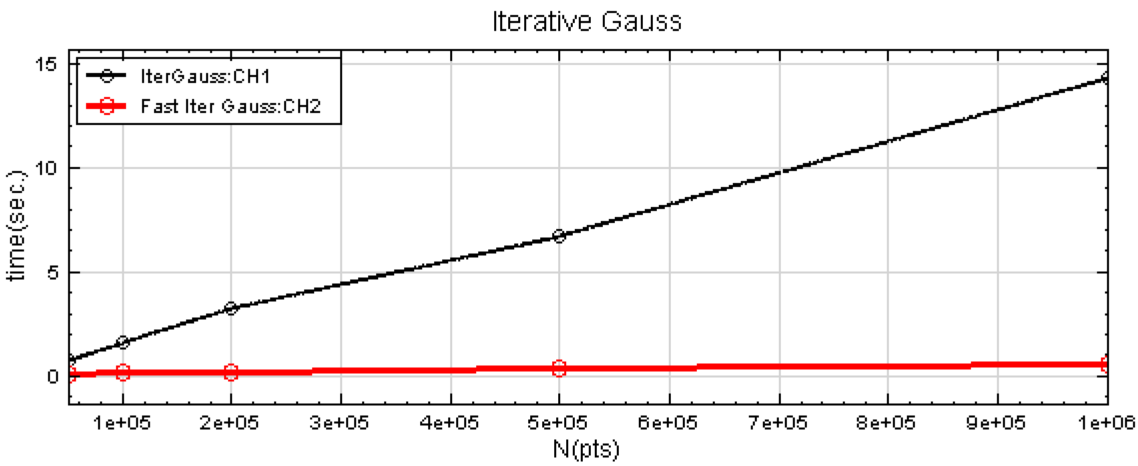
This module accepts real numbers, single channels, regular signals and audio inputs; the format of the input signal is the plural and single channel spectra data. Properties are set up as the below table:



Property Name	Property Definition	Default Value
Filter Type	The type of the Iterative Gaussian Filter: LowPass：extracting the low frequency part of the signal. HighPass：extracting the high frequency part of the signal. ByPass：reserving all frequency parts of the signal.	LowPass
Attenuation	The property of the Gaussian curve of the filter.	0.01
FH	Filtering higher values of the signal than the FH.	10
FL	Filtering lower values of the singal than the FL.	2

Example

Based on the computer with the CPU 2.8GHz (intel E6300), 3. 25GRAM, points by computing time figure for computing signals with Brown noises of different lengths by the standard/fast version Iterative Gaussian Filter is shown as below:



More important, the standard Iterative Gaussian Filter can just process 2 million point data with the 3.25G memory; the fast version can process 160 million point data.

Related instructions

Trend Estimater，Iterative Gaussian Filter。

References

Yih Nen Jeng, "Diffusive and Fast Filter Using Iterative Gaussian Smoothing", Department of Aeronautics and Astronautics, National Cheng Kung University

YIH-Nen Jeng P. G. Huang You-Chi Cheng, 2007, "Decomposition of one-dimensional waveform using iterative Gaussian diffusive filtering methods", Proc. R. Soc. A doi:10.1098/rspa

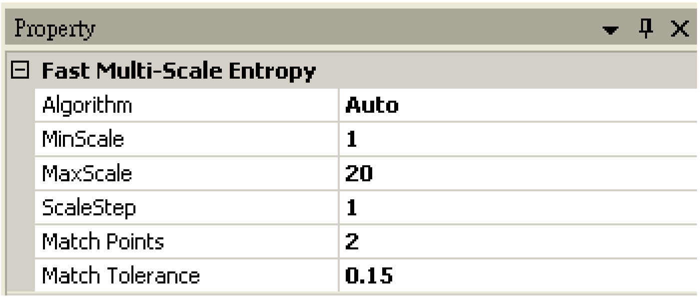
Yih-Nen Jeng, You-Chi Cheng, 2006, "Accuracy Comparison between Two Sharp and Diffusive Filters", Proc. R. Soc. A doi:10.1098/rspa

3.8.5 Fast MSE

In the algorithm of the standard MSE, the time for computing a Scale is proportional to the sqare of the length of the signal.

Properties

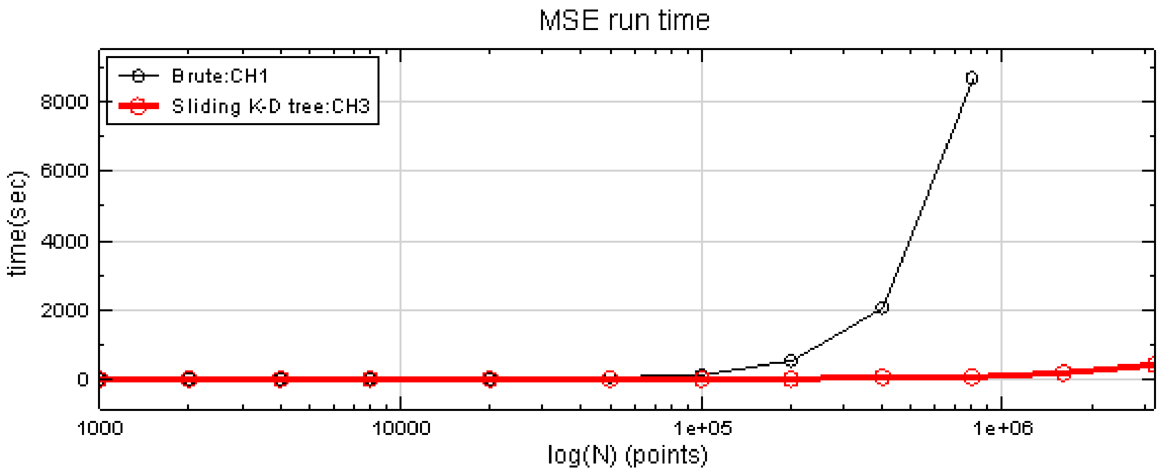
This module accepts real numbers, single channels, regular signals and audio inputs; the format of the input signal is the plural and single channel spectra data. Properties are set up as the below table:



Property Name	Property Definition	Default Value
Algorithm	The options of the algorithm include the Auto, the Brute, the Sort and the K-D tree. The Brute is the algorithm of the MSE. When the length of the signal is larger than 5000 points, the K-D tree is chosen, otherwise the Sort is chosen.	Auto
MinScale	The minimum scale is the scale's lower limit of the dimensional analysis.	1
MaxScale	The maximum scale is the scale's upper limit of the dimensional analysis.	20
ScaleStep	The increasing/decreasing step of the scale.	1
MatchPoint	Setting the series length of judging the similarity.	2
MatchTolerance	Setting the tolerance of judging the similarity.	0.15

Example

Based on a computer (Intel dual core E6300 2.8 GHz), the graph of computing time vs data points for both the standard and fast MSE is shown as below. The test signal is the pink noise, other properties are all default values.



Related instructions

Noise, Viewer.

References

1. Pincus, S. M., Approximate entropy as a measure of systemcomplexity, Proceedings of the National Academy of Sciences, USA, Vol. 88, pp. 2297-2301 (1991).

2. Costa M., Goldberger A.L., Peng C.-K. Multiscale entropy analysis of physiologic time series. Phys Rev Lett 2002; 89:062102.

3. Costa M, Peng C-K, Goldberger AL, Hausdorff JM.Multiscale entropy analysis of human gait dynaiics. Physica A , 2003;330:53-60.

4. Costa M., Goldberger A.L., Peng C.-K. Multiscale entropy analysis of biological signals. Phys Rev E 2005;71:021906.

3.8.6 Peak Detection

The Peak Detection can be used for intercepting the position of the peak signal or calculating the time difference of the two peak signals.

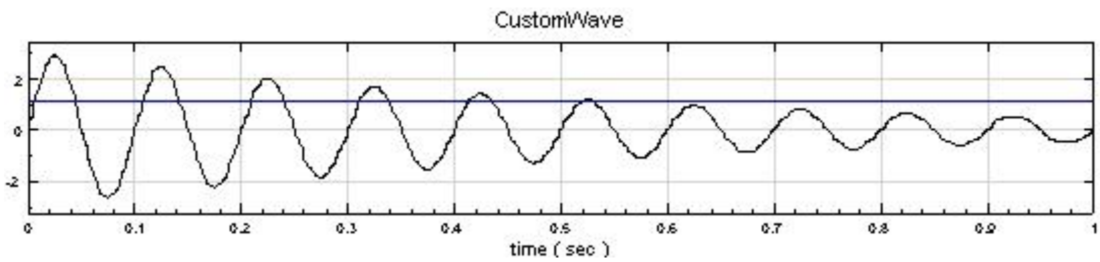
Introduction

(1) The peak is defined as the maximum value in one cycle. This maximum value is the mathematical Local Maximum.

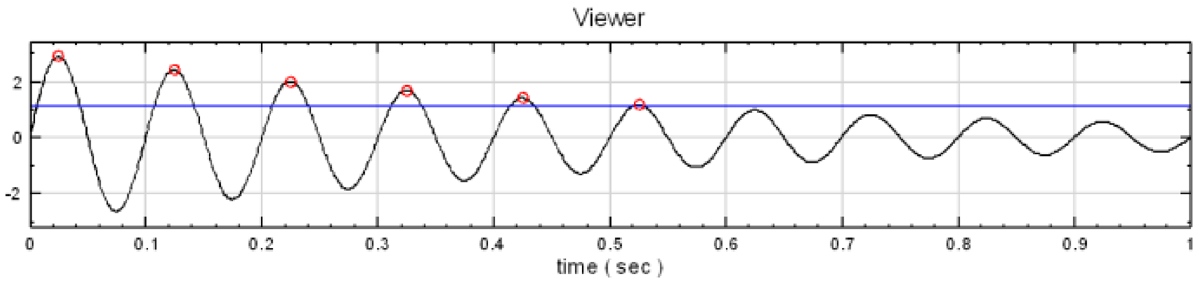
(2) The judgement could be disturbed by noises. To eliminate this disturbance, the user could firstly filter the signal by the FIR, the IIR or the EMD, and then intercept the peak.

In this module, the differentiation is used to enhance the gradient of the signal so that the maximum is more obvious in this area. Here the EMD is the filter. If the user would like to intercept the higher frequency peak, the EMD could separate the useful data from the signal. If the Speckle Noise is mixed with a signal, the EMD could separate the noise from the signal.

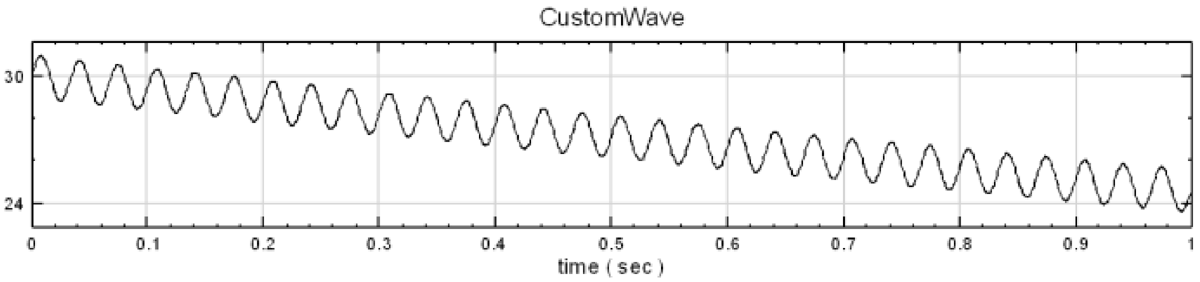
To use this module, the user should define the threshold firstly:



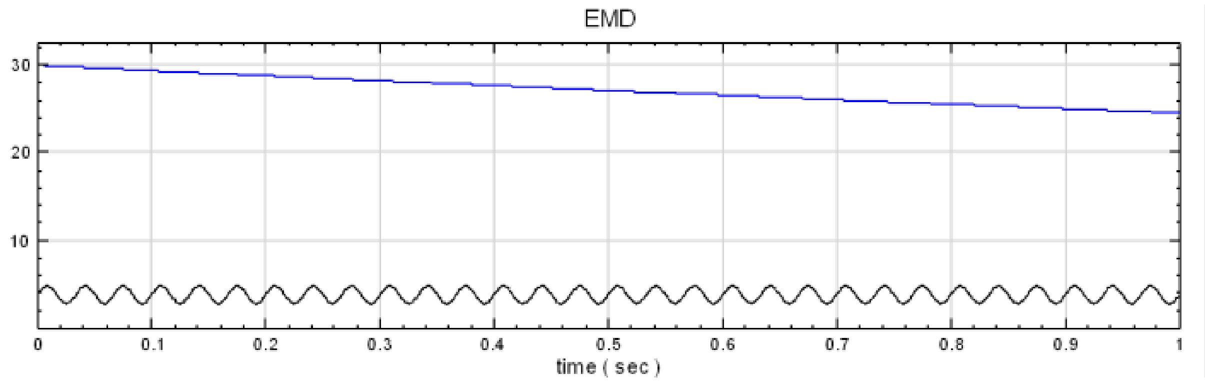
The peak above this threshold is intercepted by the Peak Detection:



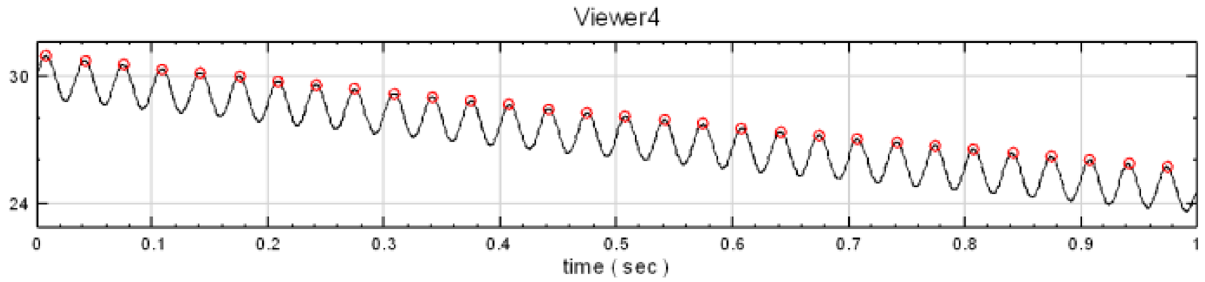
If the signal is in the trend pattern, as below:



The EMD could be used for filtering the trend:



And then, the peak is intercepted:



Properties

This module accepts real numbers, single channels, regular signals and audio inputs; the format of the input signal is the plural and single channel spectra data. Properties are set up as the below table:

Property	
EMD Stop Criteria	
StandardDeviation	**0.1**
Max. Sifting Iterations	**10**
Module	
Peak Detection	
Differential	**False**
Output Type	**PeakVsTime**
Relative Threshhold	**True**
Threshhold Ratio	**0.4**
Use EMD	**True**
Target Frequency	**40**

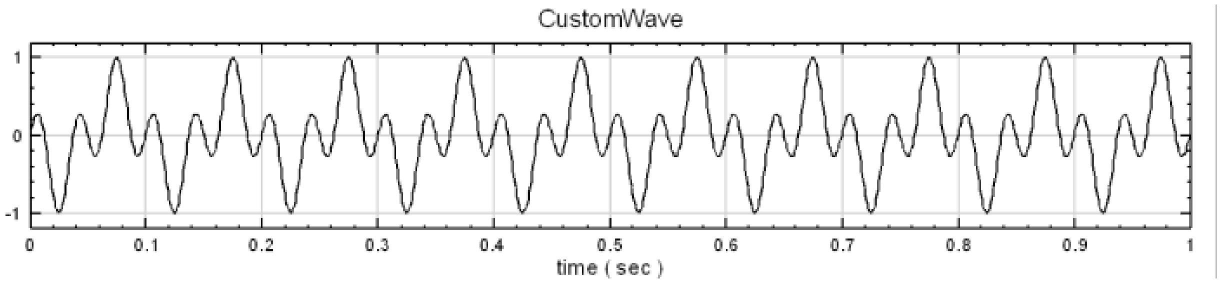
Property Name	Property Definition	Default Value
Differential	Judging whether the peak threshold is the square of the differentiation of the signal.	False
Output Type	The type of the output could be the Peak vs Time	PeakVsTime

	or the Peak To Peak Interval.	
Relative Threshold	Whether the relative threshold or the absolute threshold is used.	True
Threshold Ratio	The ratio of the benchmark of the peak and the maximum value (0.0~1.0).	0.4
Threshold value	The user difined threshold	0.4
Using EMD	Whether the EMD is used for filtering the trend and the high frequency part of the signal.	False
Target Frequency	The signal above this target is romoved.	40
StandardDeviation	Judging wether the IMF is convergent or not.	0.1
Max. Sifting Iterations	The maximum iteration time of the IMF.	10

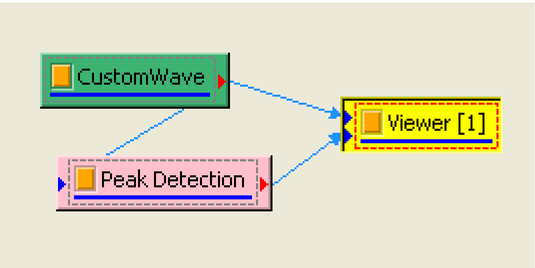
Example

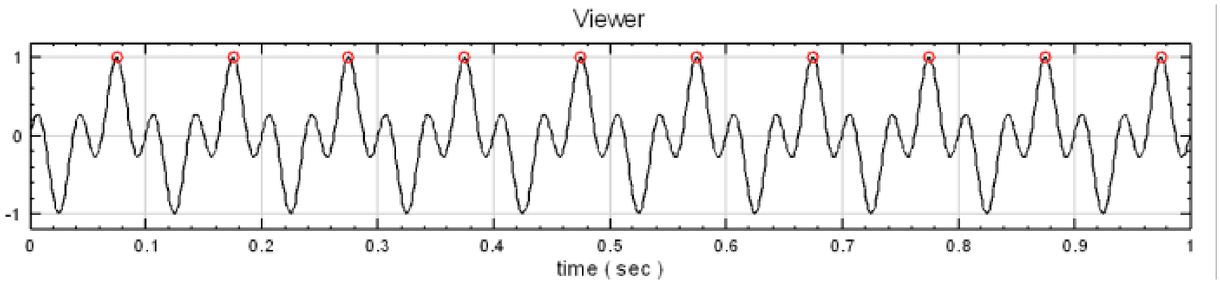
the wave form of the manual signal:

Generating the wave by the Source / Custom Wave with default properties, the Expression is sin(2*pi*10*t)*cos(2*pi*20*t). And the figure is shown as below:

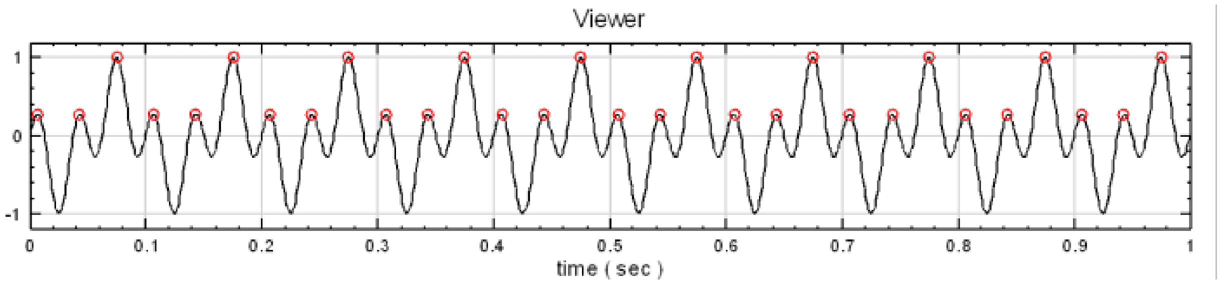


And then link the wave to the Peak Detection with default properties. The result is viewed by the original Viewer.

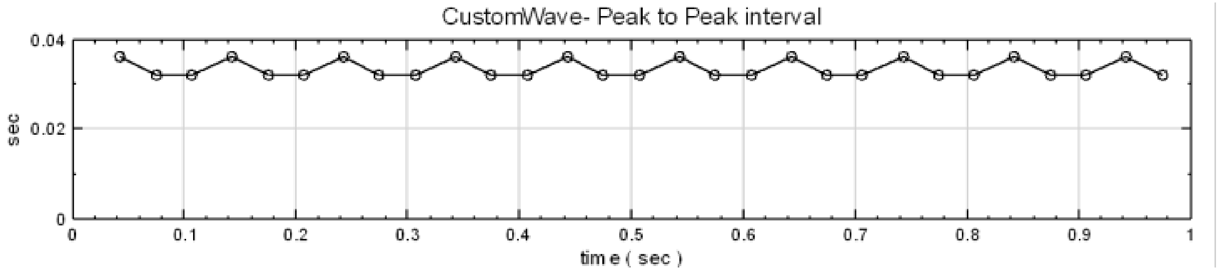




Setting the threshold to be 0.2:

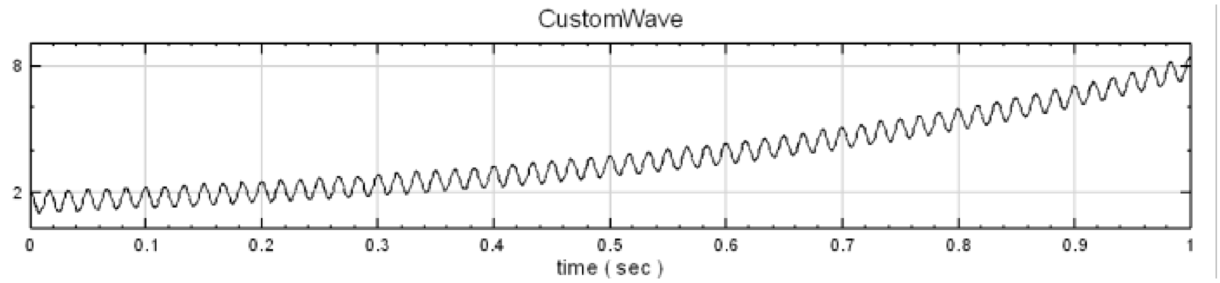


The PeaktoPeakInterval is shown as below: where the time axis presents the time of the peak and the vertical axis presents the time difference of the peak and the last peak.

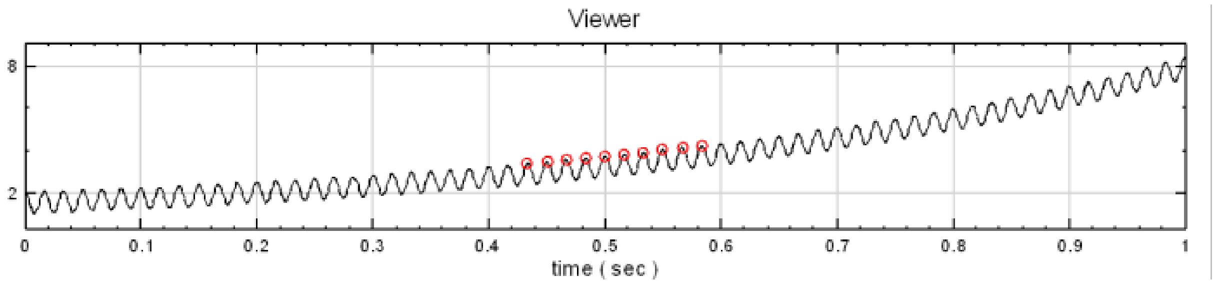


the manual signal with the trend pattern:

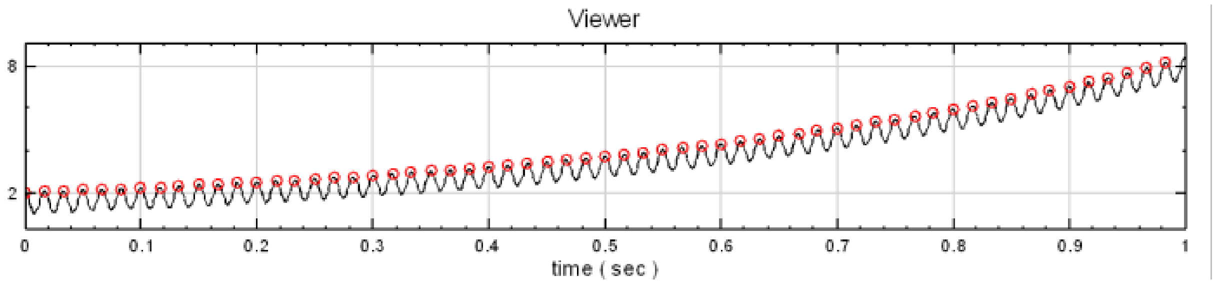
Generating the wave by the Source / Custom Wave with default properties, the Expression is cos(2*pi*30*t)*cos(2*pi*30*t)+exp(2*t). The figure is shown as below:



The correct peak can not be intercepted with the default properties of the Peak Detection.

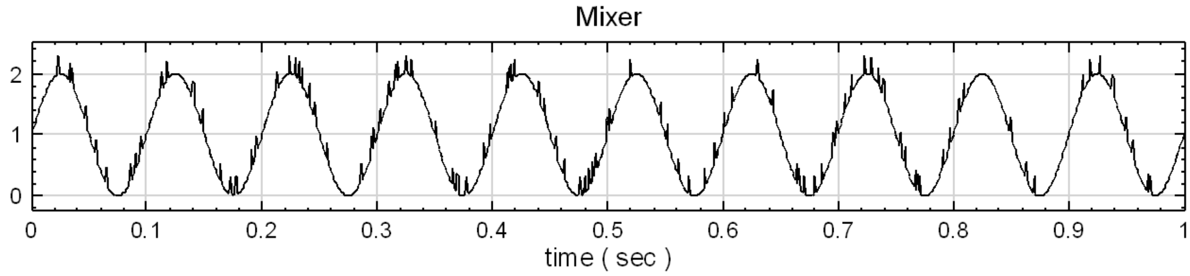


Turning on the EMD, setting the Threshold Ratio to be 0.6 and the TargetFrequency to be 70, the result is shown as below:

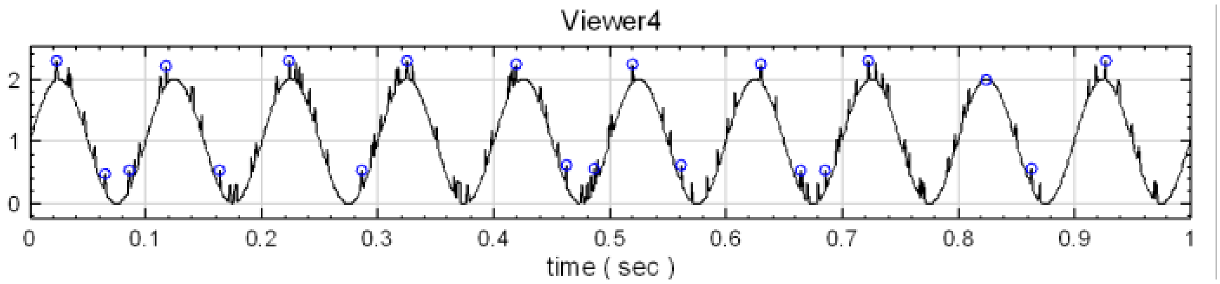


the manual singal with the Speckle Noise:

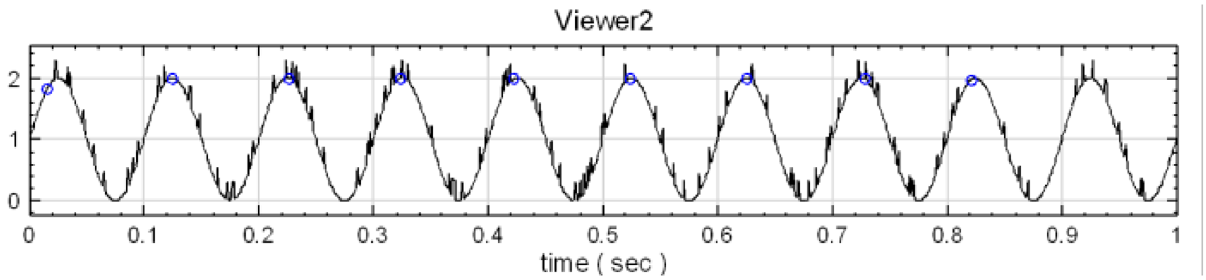
Mixing the Sine Wave (frequency=10) with the positive Speckle Noise, whose amplitude is the 30% of the Sine wave; the mixed wave is shown as below:



Intercepting by the Peak Detection with the default properties, the result is shown as below:



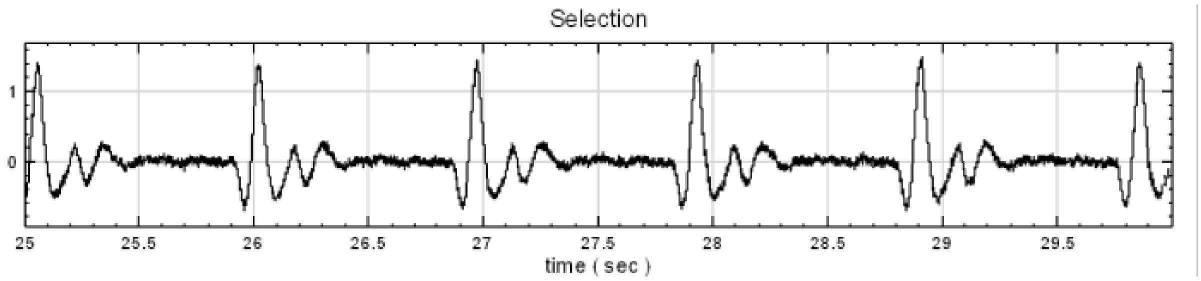
Turning on the EMD, reserving the value that is lower than frequency=10, the Target Frequency is set to be 12. The result is shown as below:



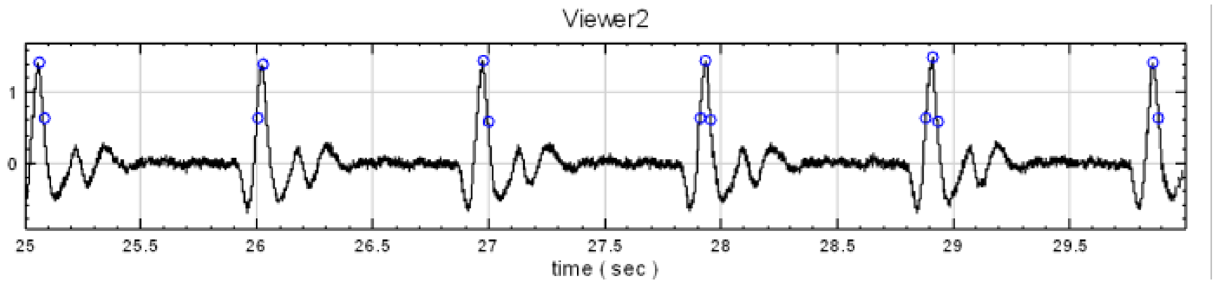
The result is better.

pulse sound diagnostic device signal:

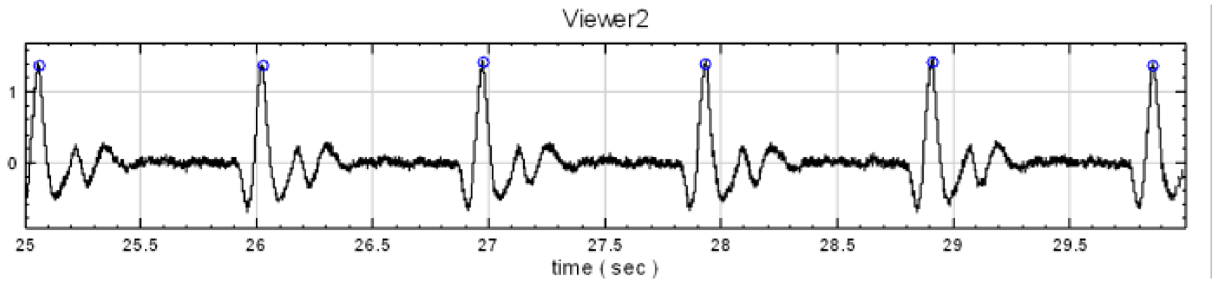
The original data is shown as below:



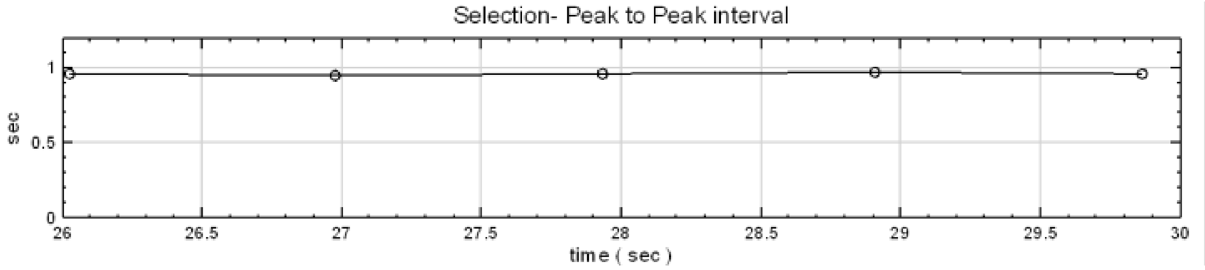
Intercepting by the Peak Detection with default properties, the result is not good.



Turning on the EMD, setting the Target Frequency to be 40 Hz, the result is better. The reason is that there is much noise in the signal, the EMD is used to filter the useful data from the signal.



The Peak to Peak Interval is:



Related instructions

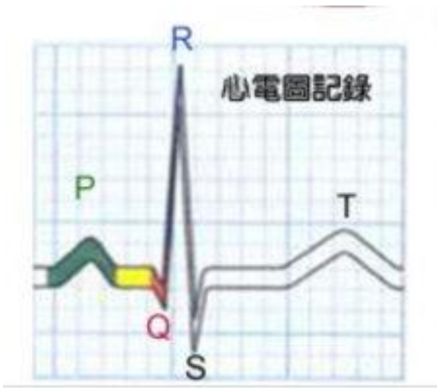
RRinterval, EMD.

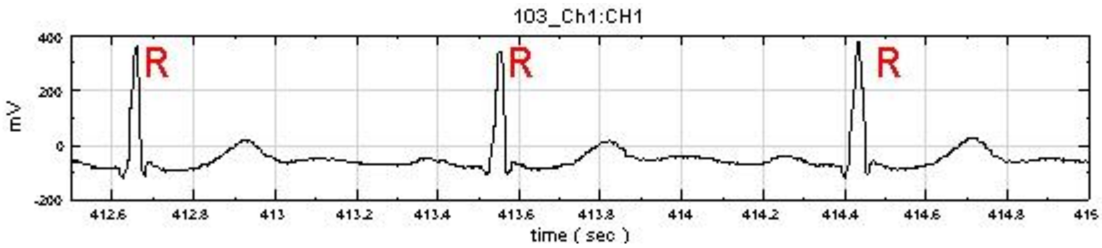
3.8.7 R-R interval

The detection of the R wave is the most issue in the ECG signal diagnosis. The R-R interval could be used to diagnose many kinds of diseases.

Introduction

The ECG signal includes P、Q、R、S and T wave types.





The R wave is usually the most obvious peak. This module intercepts the time interval of two R waves. In the output figure, the x axis presents the time of the R wave; the y axis presents the time difference of two R waves.

Properties

This module accepts standard ECG signals, the D-value of voltages of two electrodes with the unit: Volt or milliVolt. The input signal is at least 3000 points in length (Actually, the first 3000 points are not used for computing, but for reference.).

This module accepts real numbers, DeltaVoltages, TwoElectrodes. The Regular type of the output data is the index DeltaVoltage. Properties are set as below:

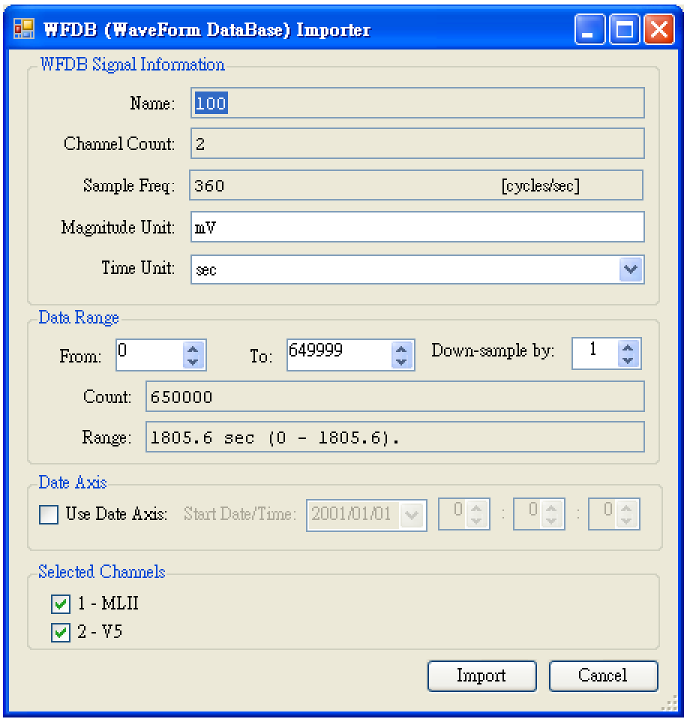
⊞ **Module**	
⊟ **R R interval**	
Type	**TwoElectrode**
Unit	**milliVolt**
Gain	**200**
DCvalue	**0**

Property Name	Property Definition	Default Value
Type	Setting the type of the input data (DeltaVoltage or TwoElectrode)	DeltaVoltage
Unit	Setting the unit of the input data (Volt or milliVolt)	milliVolt
Gain	The ratio of changing the ADC unit to the physical unit.	200
DCvalue	The DC value of the ECG reference value.	0

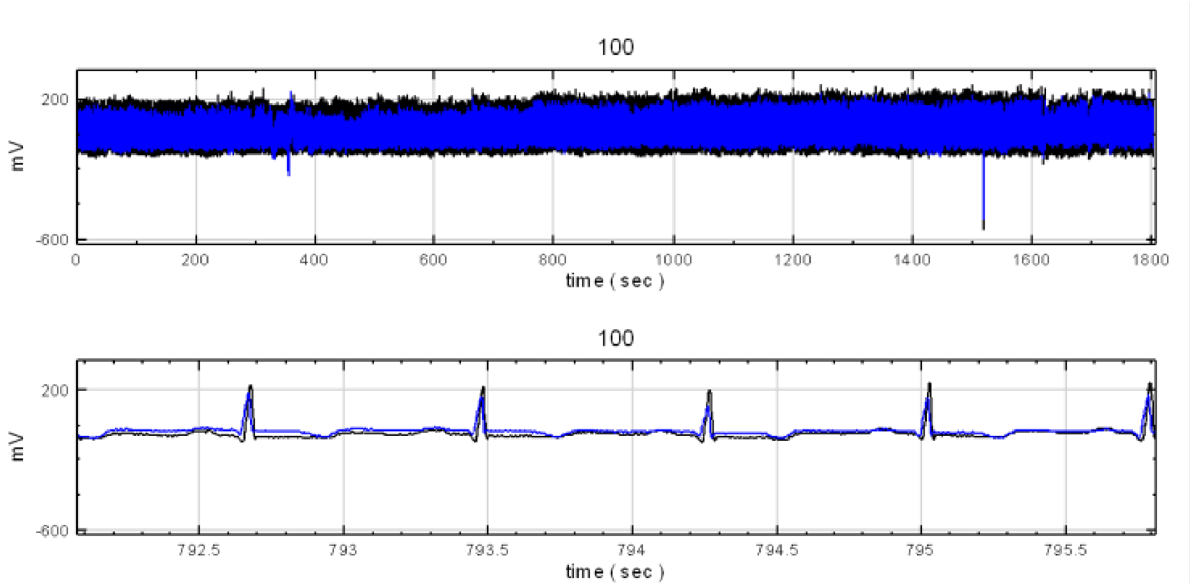
Example

taking the 100 MITDB (http://www.physionet.org/physiobank/database/mitdb/) for example:

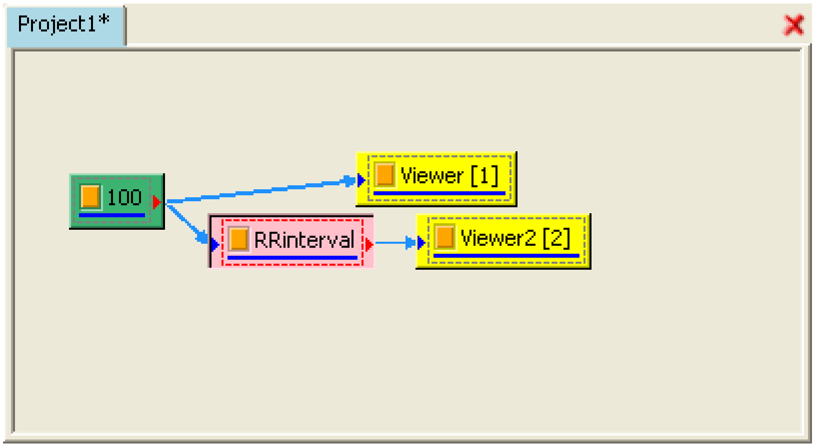
The format of the input data is a hea file. When the data is imported, we can find that the data is a TwoElectrode signal with the unit of mV:

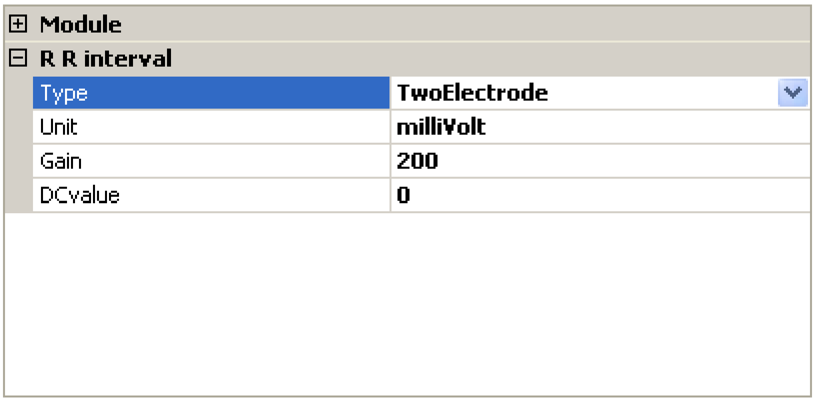


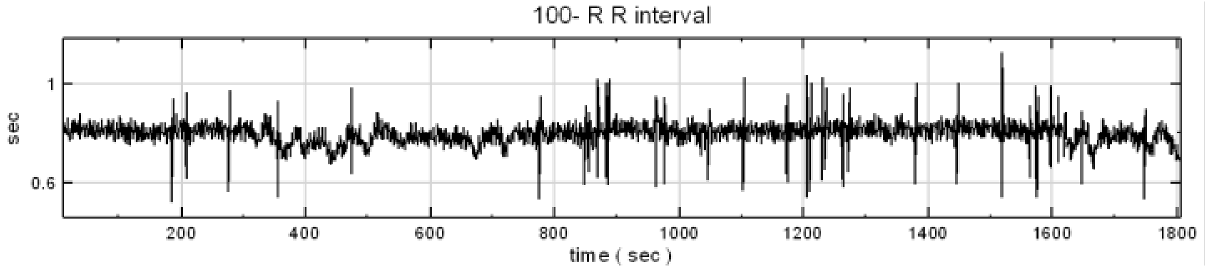
The result is viewed with the Channel Viewer:



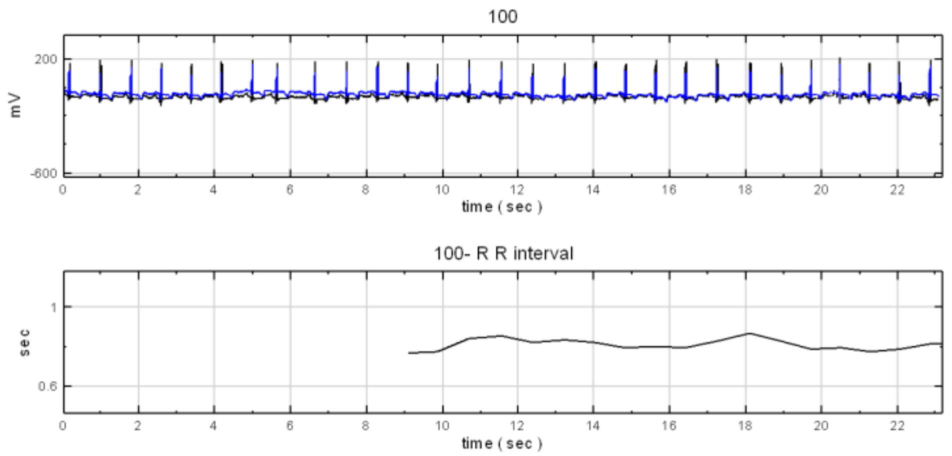
And then, linking the singal to the RR interval to process the signal and setting the Type to be the TwoElectrode:



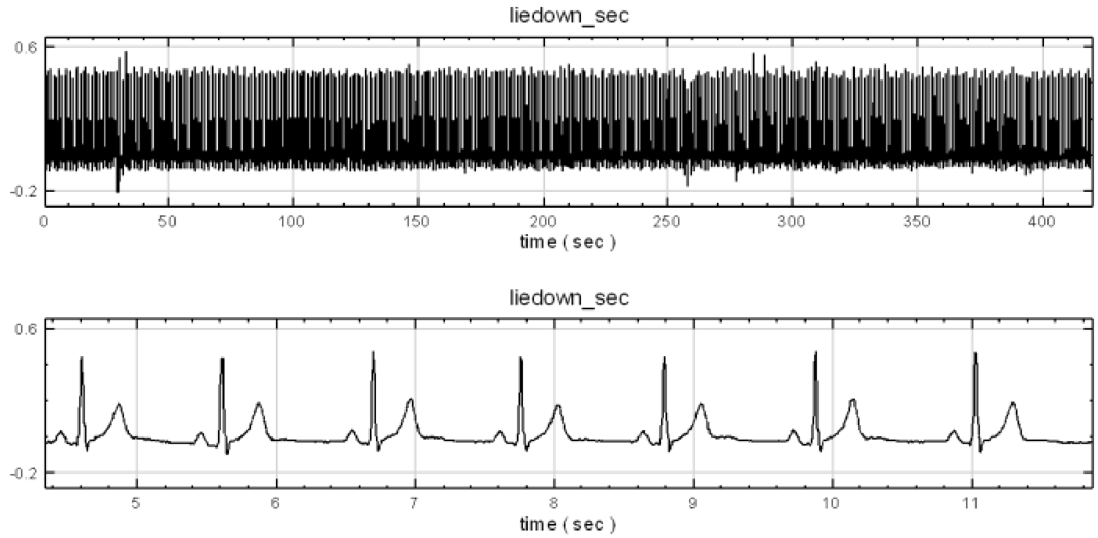




Because the first 3000 points of the data are used for reference, the start time of the RR interval is at the 9.1 second.

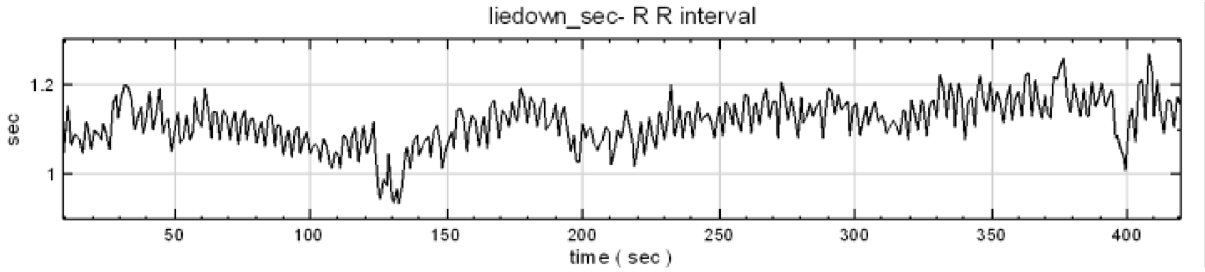


The ECG signal is collected from a subject lying on a bed. It is the voltage between two electrodes and the unit is Volt.



⊞ **Module**	
⊟ **R R interval**	
Type	**DeltaVoltage**
Unit	**Volt**
Gain	**200**
DCvalue	**0**

And then linking the signal to the RR interval to process the data and setting the type to be the DeltaVoltage, the result is shown as below:



This module is very sensitive to the properties of the input signal. If the input signal is not the standard ECG signal, the module presents a warning and may not be able to finish the computation.

The user could make the RR interval process for the nonstandard ECG signal by the Peak Detection module.

Related instruction

Peak Detection.

3.8.8 Teager

The Teager Energy Operator is a nonlinear differential operator based on the time-frequency product. It could be used for analyze the modulation of the signal. Based on the modulation, the instantaneous frequency and amplitude are defined.

Introduction

Based on the AM-FM model, the signal could be rewrited in the following way:

$$x(t)=a(t)\cdot cos(\Omega_c t+\Omega_m\int_0^t q(m)dm+\theta)$$

Where Ω_c is the Carrier frequency, $q(t)$ is the Information signal $|q(t)|<1$, Ω_m is the maximum frequency offset $0<\Omega_m<\Omega_c$, $a(t)$ is the instantaneous amplitude, θ is the initial phase angle.

Defining the Teager Energy Operator as:

$$\Psi_d[x(t)]=\frac{[x^2(t)-x(t-\Delta t)\cdot x(t+\Delta t)]}{T^2}$$, where T is the period of sampling.

Based on the linear narrow modulation, the upper formula is approximated as below:

$$\Psi[a(t)\cdot cos(\Omega_c t+\Omega_m\int_0^t q(m)dm+\theta))]\approx a^2(t)\cdot sin^2(\Omega_i(t))$$

Defining $y(t) = x(t)-x(t-\Delta t)$ (backward difference)

And then:

$$\frac{1}{2}(\Psi[y(t)]+\Psi[x(t)])\approx 4a^2(t)\cdot sin^2[\frac{\Omega_i(t)}{2}]\cdot sin^2[\Omega_i(t)]$$

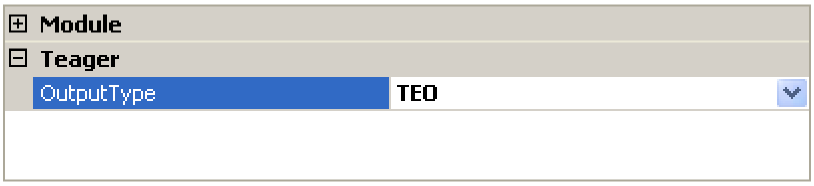
$$\Omega_i(t)=acos(1-\frac{\Psi[y(t)]+\Psi[y(t+\Delta t)]}{4\cdot\Psi[x(t)]})$$

$$|a(t)| = \left\langle \frac{\Psi[x(t)]}{\frac{\Psi[y(t)]+\Psi[y(t+\Delta t)]}{4\cdot\Psi[x(t)]}} \right\rangle$$

These are the instantaneous frequency and amplitude.

Properties

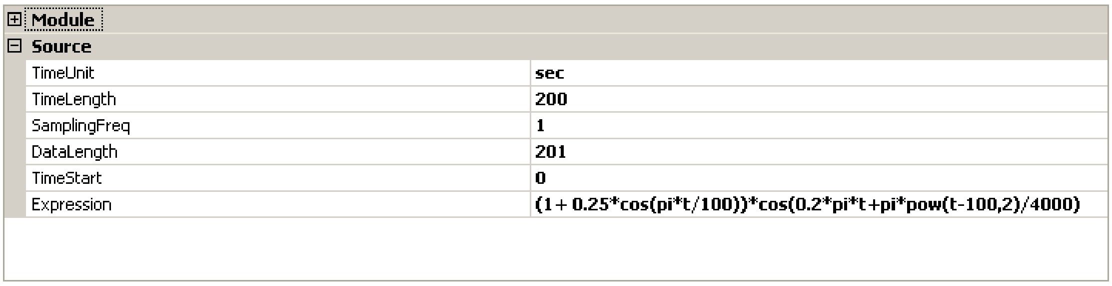
This module accepts real numbers, single channel signals and multi-channel signals. The Regular signal or the audio input has the same format of the input/output data.

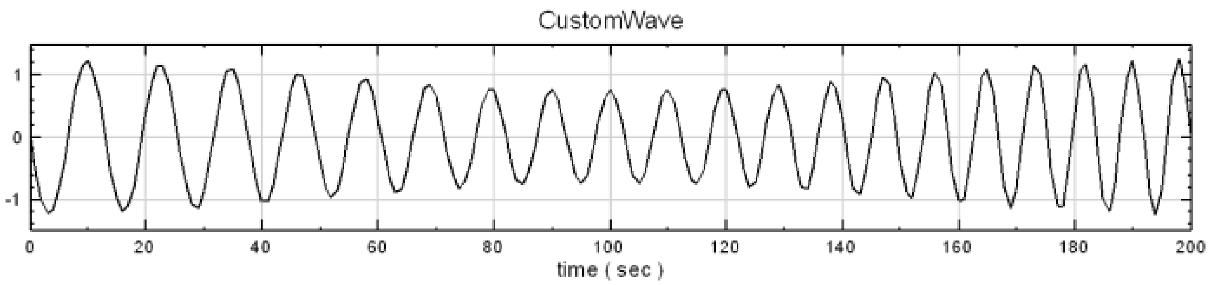


Property Name	Property Definition	Default Value
OutputType	Setting the type of the output: TEO (Teager Energy Operator), InstantAmplitude or InstantFrequency.	TEO

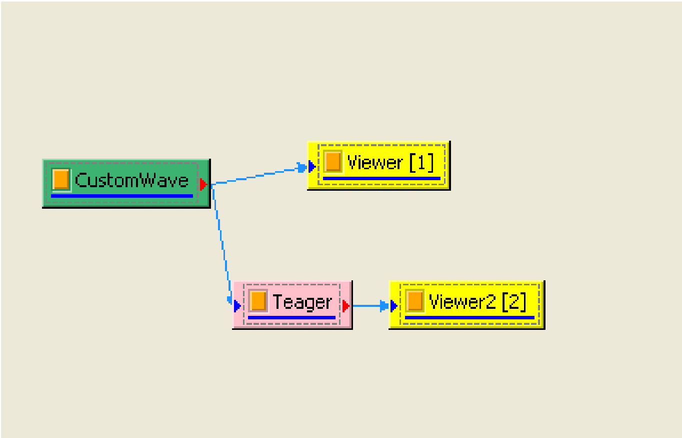
Example

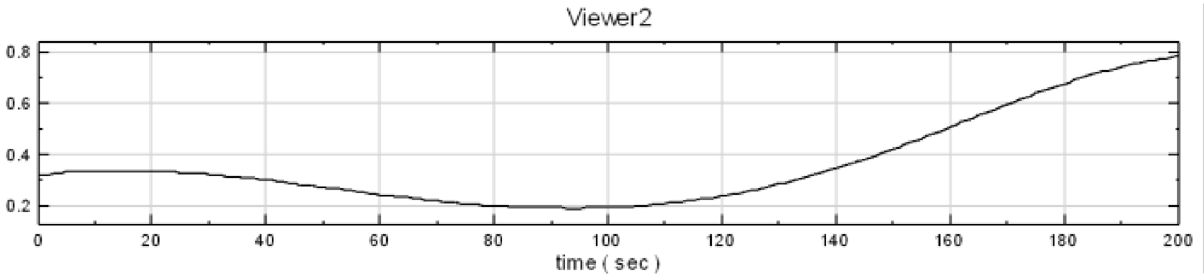
Generating the signal with the Source / Custom wave, setting the sampling rate to be 1 and the TimeLength to be 200, the Expression is: (1+0.25*cos (pi*t/100))*cos (0.2*pi*t+pi*POW (t-100, 2)/4000).



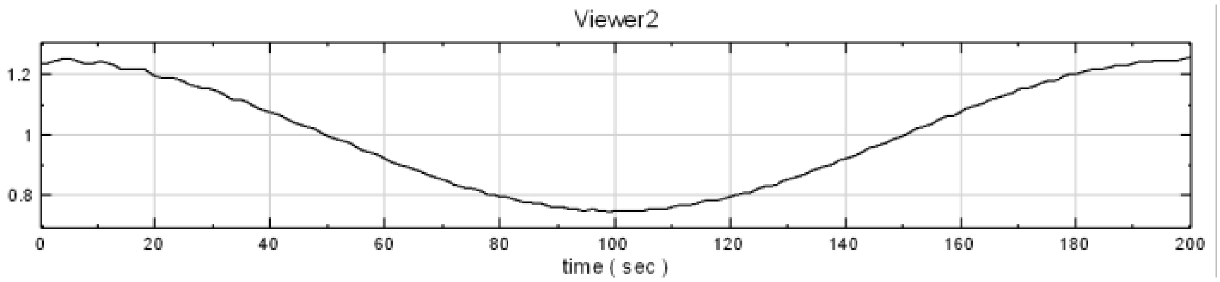


Analyzing the signal with the Teager module, setting the output type to be the TEO, the result is viewed by the Channel Viewer as below:

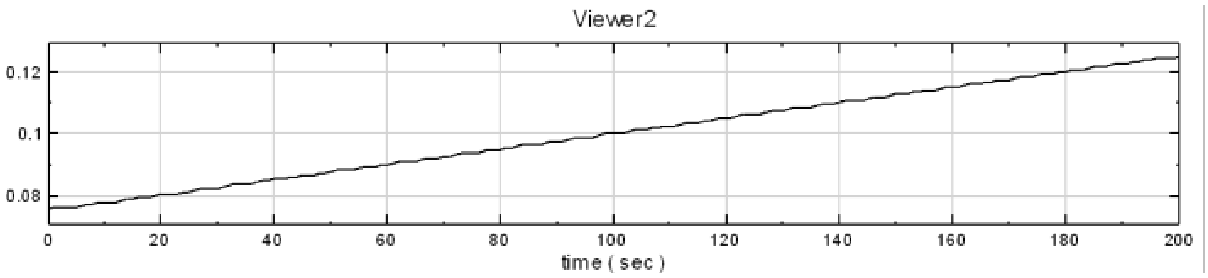




If the output type is the InstantAmplitude, the result is viewed by the Channel Viewer as below:



If the output type is the InstanceFrequency, the result is viewed by the Channel Viewer as below:

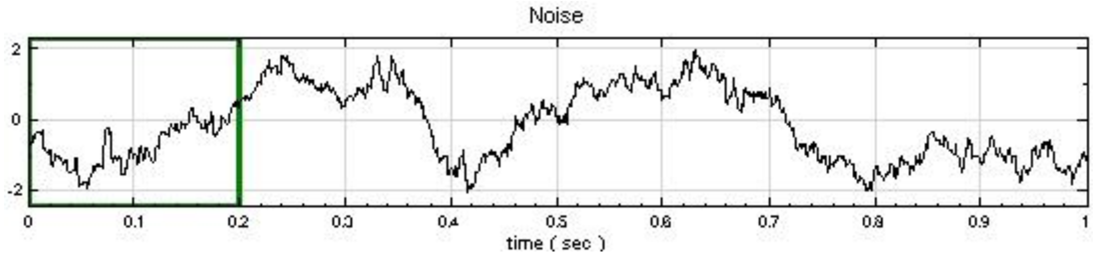


3.8.9 Rolling MSE

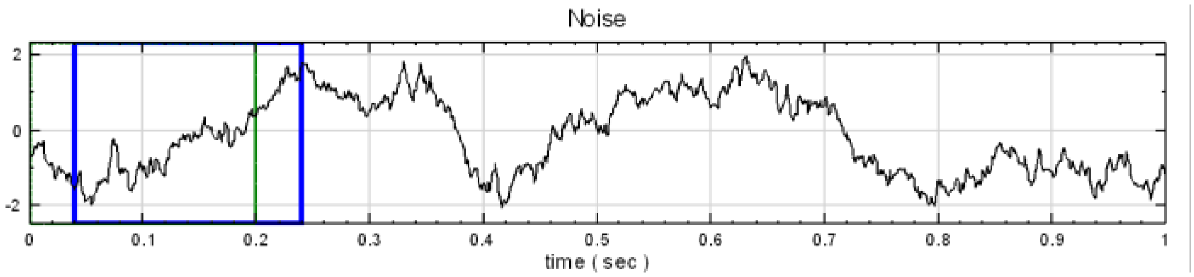
The Rolling MSE is the analysis method that makes the MSE computation for the signal, transplates the watch window of the MSE to repeat this computation and draws 3D Time-ScaleEntropy Plot. As the Fast MSE, the Rolling MSE provides three algorithms: Brute (the original algorithm, time comsumption in one dimension is $O(M^2)$), Sort (as the $O(M^2)$, but with less scale coefficients) and SlidingKD tree ($O(M^{\frac{3}{2}})$, but with more memory)

Introduction

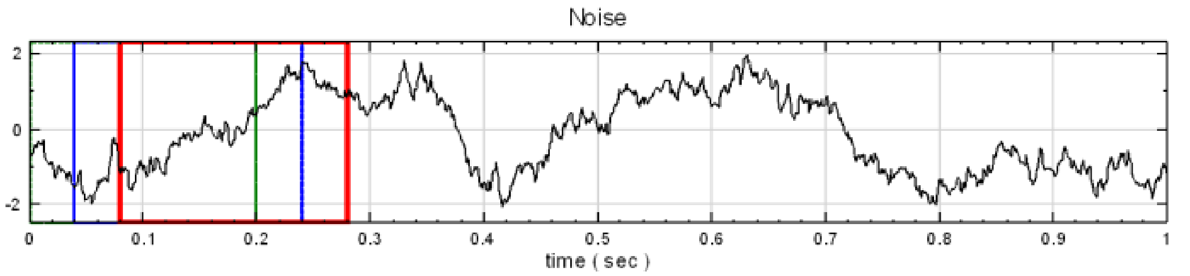
At first, extracting a section of the original data and making the MSE computation (t = 0, window length = 0.2):



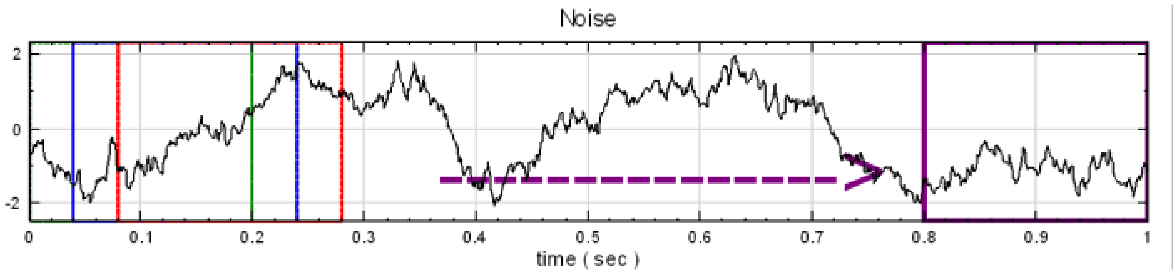
And then translating the watch window towards the right, making the MSE computation (t =0.04, the overlap is 160 points):



Translating the watch window again and making the MSE computation:

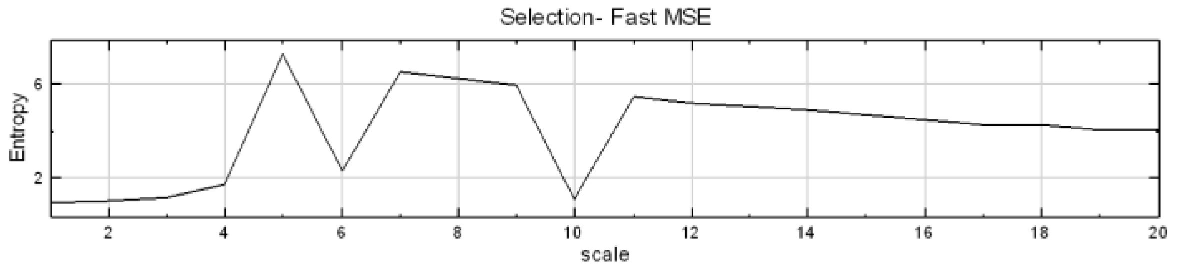


Until the watch window sweeps over the entire signal:

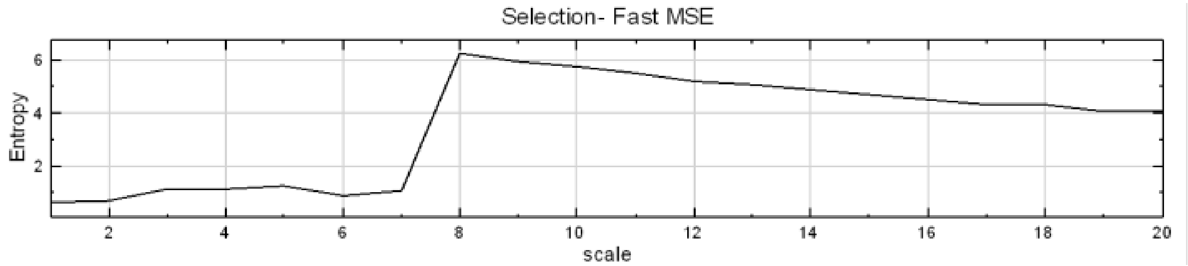


Assembling the results of all of the watch windows to be a 3D figure, the figure is shown as below:

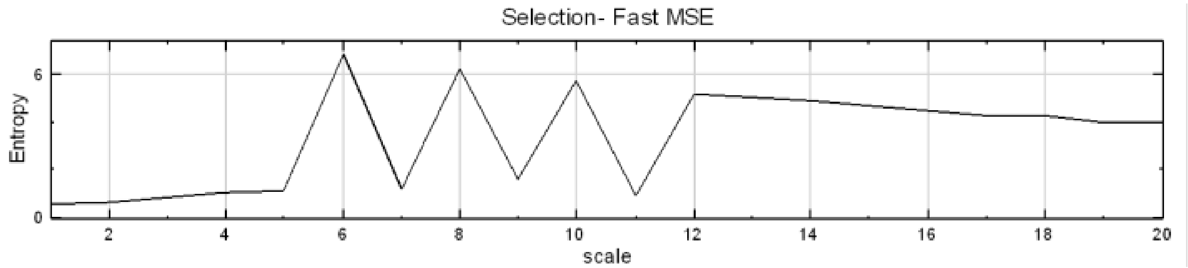
t = 0



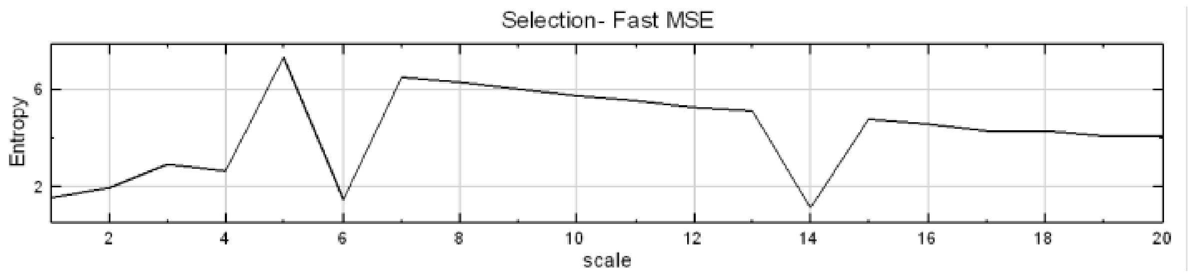
t = 0.04



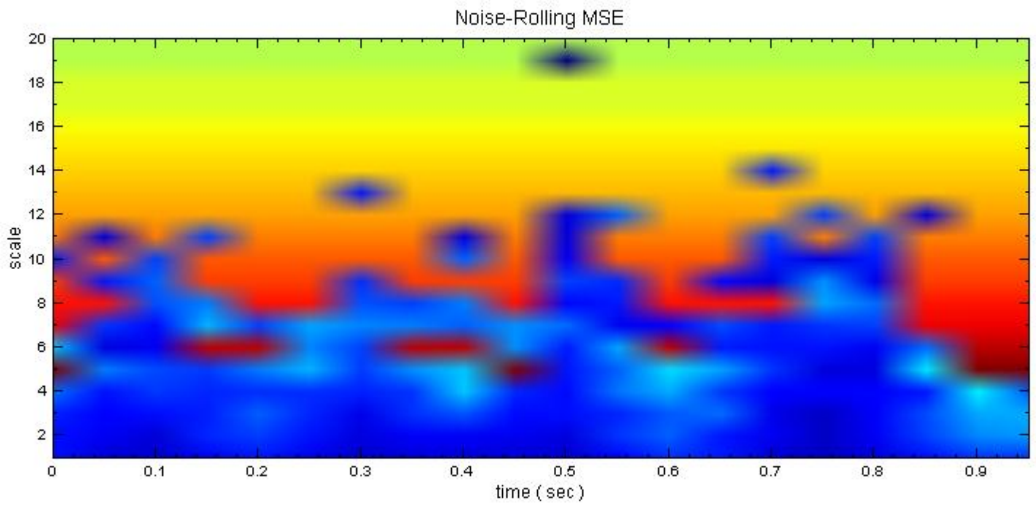
t = 0.08



t = 0.8



Arranging the results in proper time order and drawing a 3D figure as below:



The algorithm requires the length of the window to be longer than 25 points and higher than 3 times of the max scale.

Properties

This module accepts real numbers, single channels and multi-channels, regular signals, audio signals; the type of the output is real numbers, single channels and multi-channels, regular signals. Properties are the same as the properties of the original MSE, but an algorithm option is added. Detailed introductions are shown as below:

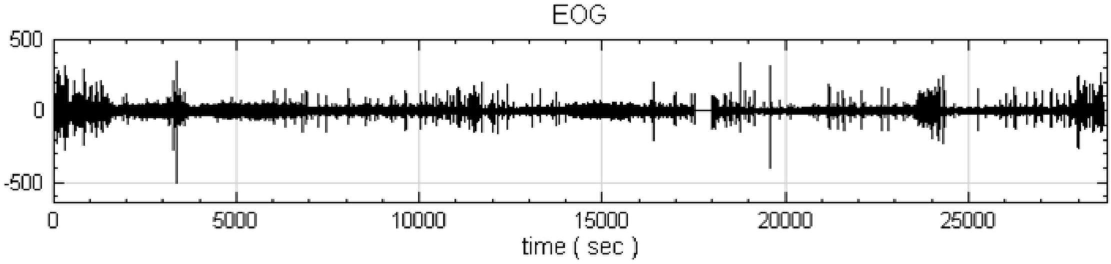
Property	
⊞ Module	
⊟ Multi-Scale Entropy	
MinScale	**1**
MaxScale	**20**
ScaleStep	**1**
Match Points	**2**
Match Tolerance	**0.15**
⊟ Rolling Multi-Scale Entropy	
Algorithm	**Auto**
Overlap	**89**
WinSize	**99**
GlobalSTD	**False**

Property Name	Property Definition	Default Value
Method	Setting the type of the algorithm: Auto, Brute, Sort and Sliding K-D Tree. The Brute is the original algorithm of the MSE; the Auto is the automatic selection. When the width of the watch is larger than 5000 points and the memory is sufficient, the Sliding K-D Tree is used, otherwise the Sort is used.	Auto
MinScale	Setting the minimum of the computation scale.	1
MaxScale	Setting the maximum of the computation scale.	20
ScaleStep	Setting the step of the computation scale.	1
MatchPoint	Setting the number of the match points.	2
MatchTolerance	Setting the tolerance of the match.	0.15
Overlap	Setting the number of the overlap points.	Depending on the length of the signal
WinSize	Setting the size of the watch window. The unit is point.	Depending on the length of the signal
GlobarSTD	Setting whether the STD of the entire signal is computed or not.	False

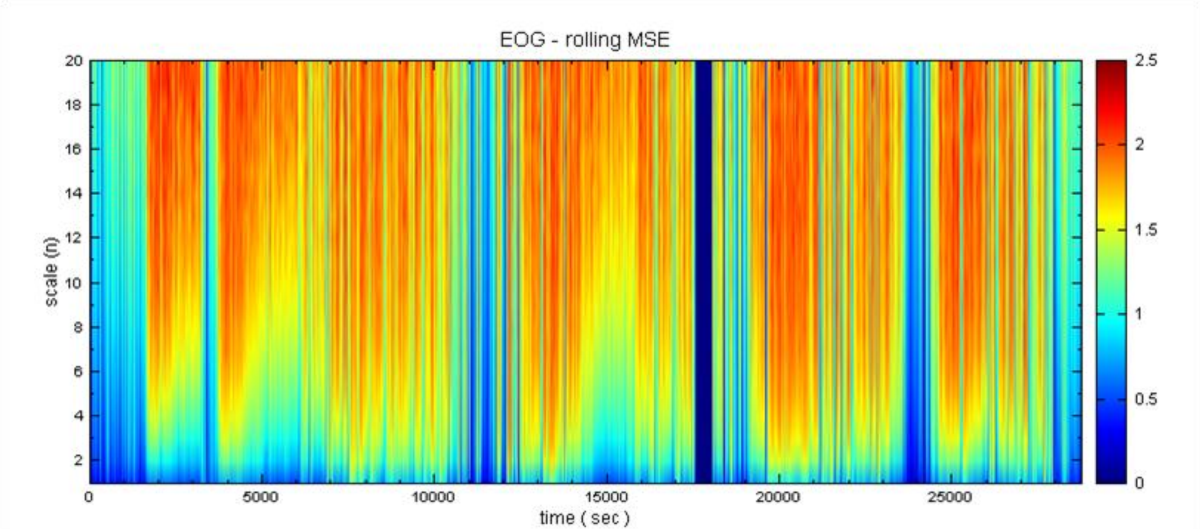
Example

The measurement of the hypnic physiological signal, including the EEG, the ECG, and the EOG etc, is shown as below.

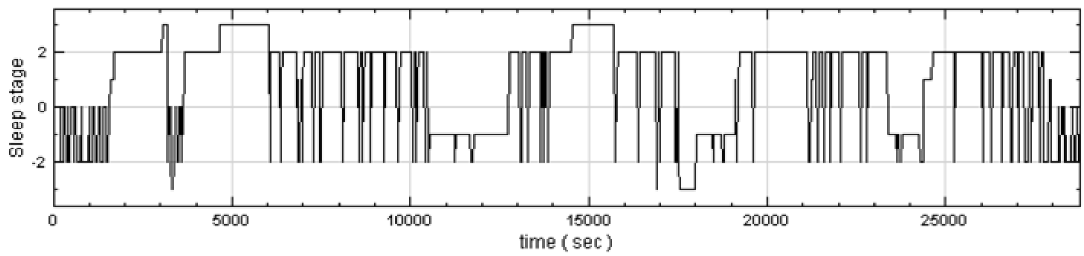
The hypnic EOG signal is shown as below:



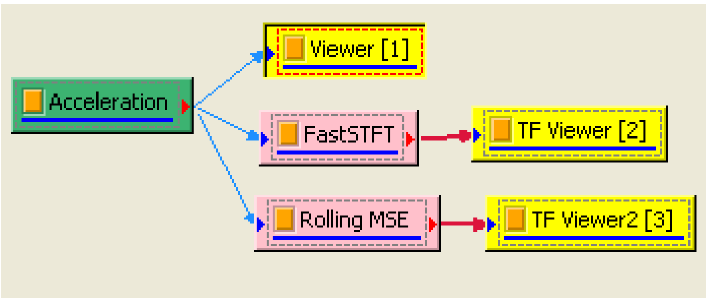
Processing the EOG signal by the Rolling MSE and watching the variation of the MSE over time:



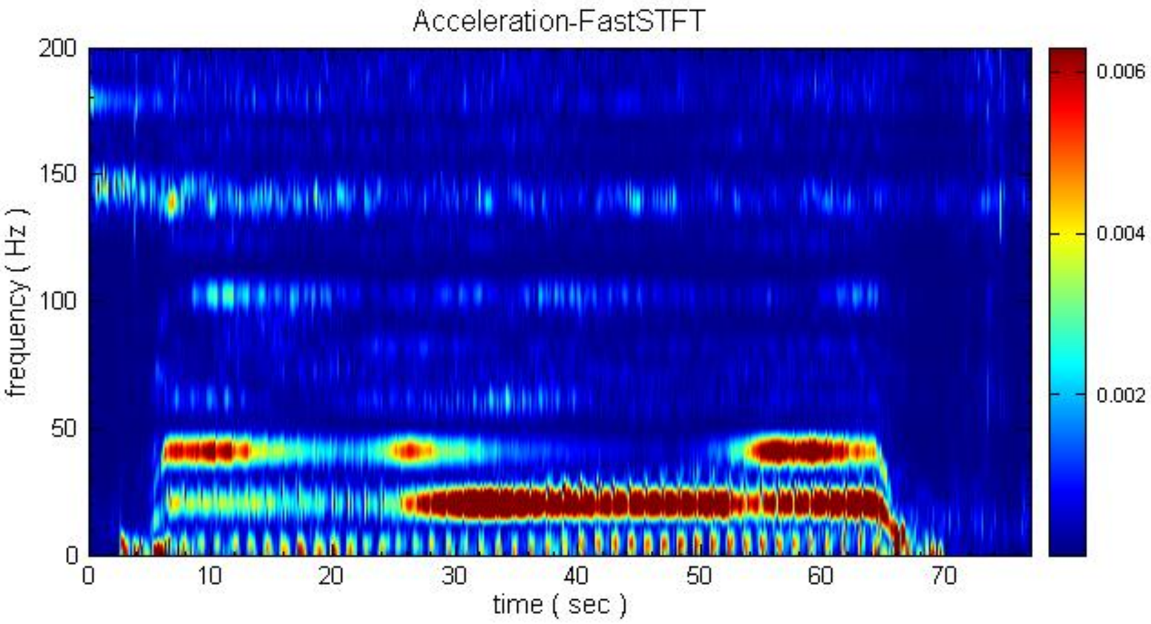
The upper figure of the Rolling MSE is similar with the lower figure of the doctor's judgement result.



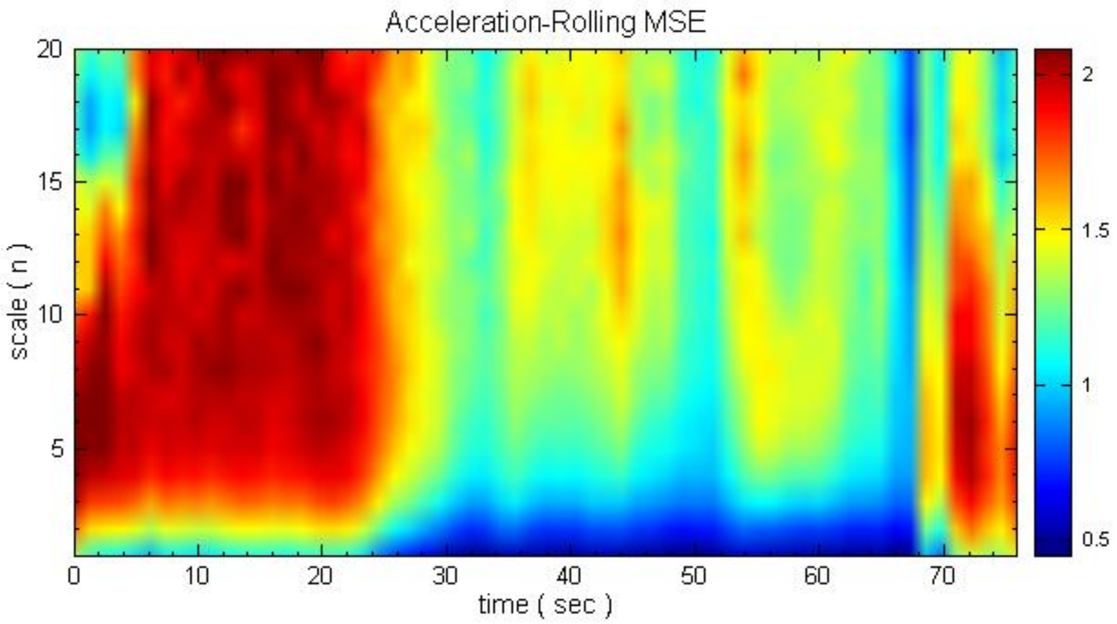
Turning on the demo73_1 (C:\Program Files \ DynaDx \ DataDemon \ demo \ Enhanced \ demo73_1 - RollingMSE.vsn), we can see an acceleration signal, which measures the vibration of the elevator from the start to the stop.



Processing the vibration signal by the Fast STFT module, the result is viewed by the TF Viewer. We can find that there is a stable 20 Hz frequency between the 25 second to the 65 second, which are the time points of the vibrations of the elevator:



And then, processing the signal by the Rolling MSE, we can find that the MSE value is lower during this time:



Related instruction

Fast MSE

References

1. L. Guzm´an-Vargas, A. Ram´ırez-Rojas, and F. Angulo-Brown：Multiscale entropy analysis of electroseismic time series；Nat. Hazards Earth Syst. Sci., 8, 855–860, 2008

2. Costa M, Peng C-K, Goldberger AL, Hausdorff JM. ： Multiscale entropy analysis of human gait dynamics. Physica A2003;330:53-60。

3. Costa M., Goldberger A.L., Peng C.-K.： Multiscale entropy analysis of biological signals. Phys Rev E 2005;71:021906.

4. Costa M., Goldberger A.L., Peng C.-K ： Multiscale entropy analysis of physiologic time series. Phys Rev Lett, 2002; 89 : 062102.

3.8.10 PCA

Introduction

The PCA (Principle Component Anaylsis) decomposes k mixing signals of X to q signals of Y, where k and q are the numbers of signals and Y are un-correlated signals. In this case, the mixing signal could be expressed by fewer signals.

Descriptions in mathematics:

We assume that the input mixing signals X include k signals and the length of the signal is n; the output signals Y include q signals and the length of the signal is n.

The purpose of the PCA is to find a matrix which meets $Y = W^T X$, where X is the mixing signals (The DC value should be removed firstly), W is named: the principle component, Y is named: the reconstructed signal. It could be certified that W is the eigenvector of the covariance matrix of X. The eigenvalue decides the contribution of each principle component to X. We could remove lesser eigenvalues to express the mixing signals by fewer signals.

NOTE:

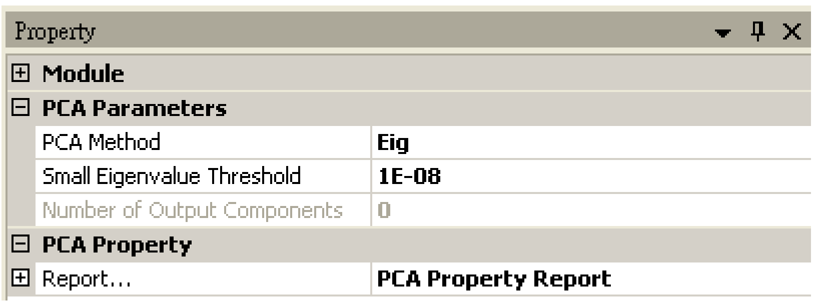
The PCA decomposes the mixing signals to un-correlated signals: the ICA decomposes the mixing signals to independent signals. The definition of the "un-correlated" is looser than the definition of the "independent".

Meanwhile, the definition of the "un-correlated" is $E(xy) = E(x)E(y)$, where E is the expected value. The definition of the "independent" is

$$\begin{aligned} & p(xy) = p(x) \cdot p(y) \\ & \Rightarrow E(g(x)g(y)) = E(g(x)) \cdot E(g(y)) \end{aligned}$$.

Properties

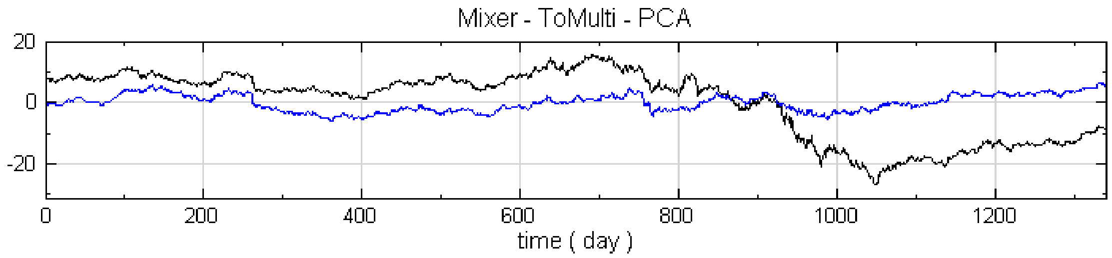
This module accepts real numbers, single channels, multi-channels, regular signals and audio signals.



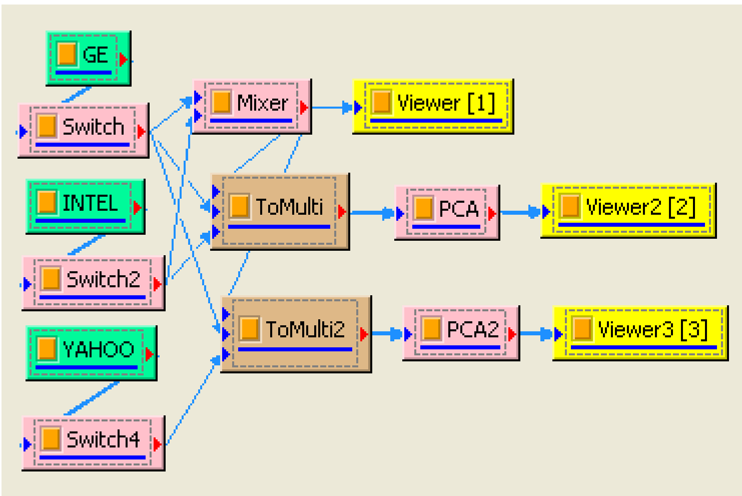
Property Name	Property Definition	Default Value
PCA method	Setting the method of the PCA: Eig (the eigenvalue method), SVD (Singular Valued Decomposition).	Eig
Small Eigenvalue Threshold	Setting the threshold below that the eigenvalue is regarded as a redundant signal.	1E-08
Number of Output Components	Setting the number of output components.	User defined
Report	Outputing the eigenvalue and the eigenvector as a report.	User defined

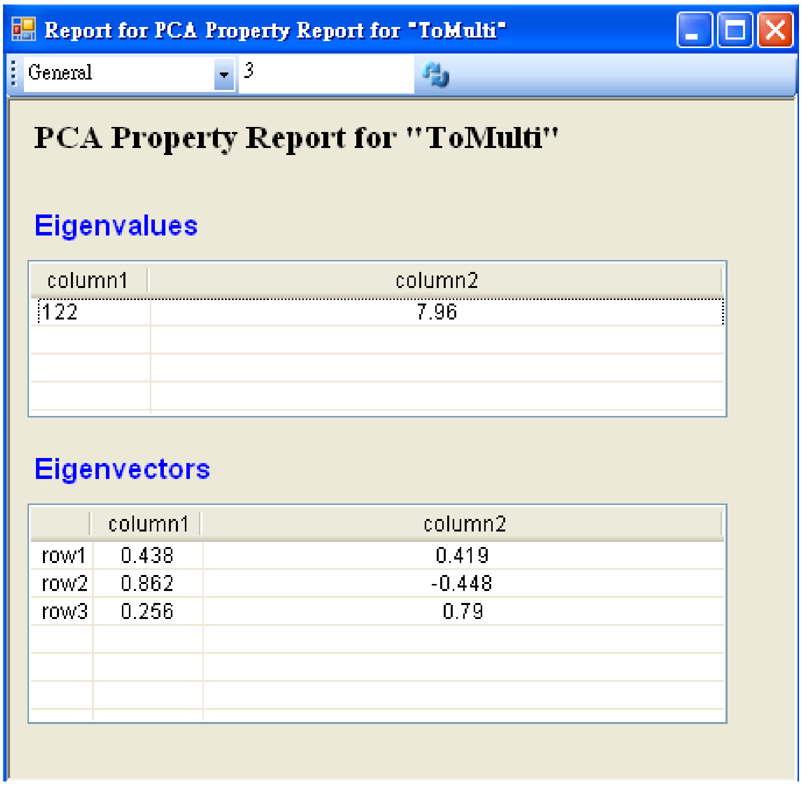
Example

Openting the example: demo 82 (C:\Program Files \ DynaDx \ DataDemon \ demo \ Enhanced \ demo 82 - PCA.vsn), the user could find three stocks: GE, INTEL, YAHOO. Mixing the GE and INTEL by the Mixer with a scale to form a foundation, the result is shown as below:



Computing the two stocks with the foundation by the PCA and stretching the report, the user could find that there are some eigenvalues tending to 0, which shows that there are GE and INTEL components in the foundation.




Report for PCA Property Report for "ToMulti"

General 3

PCA Property Report for "ToMulti"

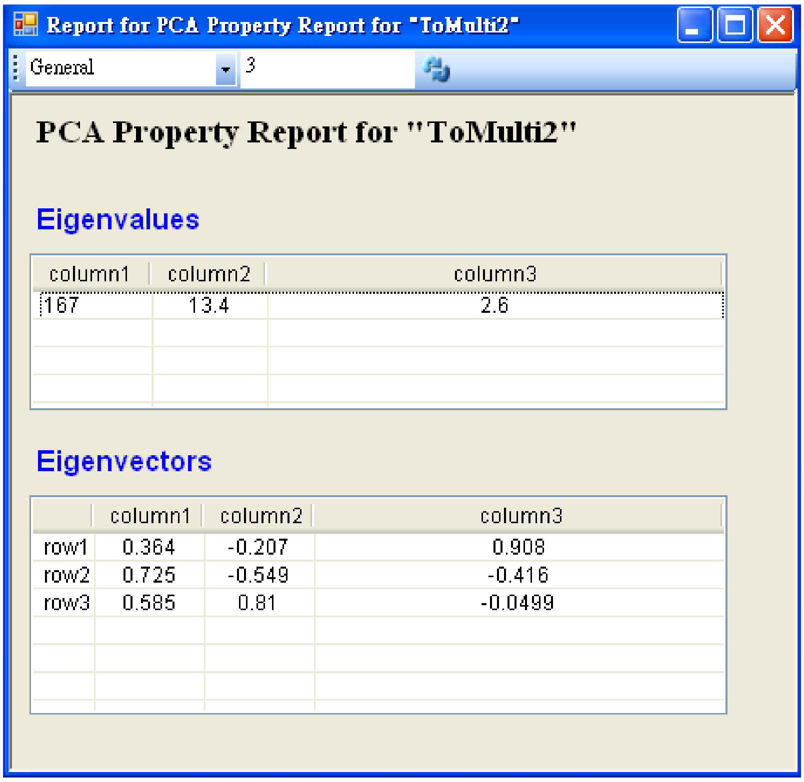
Eigenvalues

column1	column2
122	7.96

Eigenvectors

	column1	column2
row1	0.438	0.419
row2	0.862	-0.448
row3	0.256	0.79

Computing GE, YAHOO with the foundation by the PCA and stretching the report, the user could find that there is no eigenvalue tending to 0, which shows that there is no YAHOO component in the foundation.


Report for PCA Property Report for "ToMulti2"

General 3

PCA Property Report for "ToMulti2"

Eigenvalues

column1	column2	column3
167	13.4	2.6

Eigenvectors

	column1	column2	column3
row1	0.364	-0.207	0.908
row2	0.725	-0.549	-0.416
row3	0.585	0.81	-0.0499

References

1. Independent Component Analysis, ATutorial Introduction Ch10,James V. Stone, A Bradford Book.

2. Independent Component Analysis, Aapo Hyvärinen, Juha Karhunen, Erkki Oja, A Wiley-Interscience Publication.

3.8.11 ICA

Introduction

The ICA is the abbreviation of the Independent Component Analysis.

Description:

The ICA decomposes a group of mixing signals M (MixingSignal) to a group of statistic independent signals S (Source Signal).

Assuming that the M has d_m number of signals with the signal length of n, the S has d_s number of signals with the signal length of n.

The ICA is to find a matrix W that satisfies:

$$S = WM$$,

or

$$M = AS \quad W = A^{-1}$$

Where A is $d_s \times d_m$, W is $d_m \times d_s$.

There are two assumptions in the ICA: S is independent signal to each other; one S at most has the Gaussian distribution.

Application: Cocktail Part Problem

The ICA could be used for decomposing the mixing sound signal: assuming the speed of the audio signal is infinite, we set d_m number of mics at different positions to receive d_s number of different sound signals. The ICA could decompose the mixing signal to the original sound signals. If the d_m > d_s, the ICA could dump noise signals to a redundant signal which improves the contrast of the signals.

Algorithm description:

This module calculates W and S by the Fixed Point. The description is shown as below:

The independence of the signal could be described by the Non-Gaussian distribution. This method is similar with calculating the zero position of the derivative of the Cost (Objective) Function. The W is caculated by the Newton interation method:

For (n=1: MaxIteration) {

$$Max(f) \rightarrow \frac{df}{dW} = 0$$

$$W_{n+1} = W_n - \frac{f(W_n)}{\frac{\Delta f}{\Delta W_n}}$$

}

(a) Computing one column of the W: Deflation Method;

Computing all columns of the W: Symmetric Method.

NOTE: the Deflation method could accumulate Round-Off errors, so the Symmetric method is used more often.

(b) The Cost Function could be the Tanh, the kurtosis (Fourth Order Moment), the Skewness (3rd Order Moment). Usually, the tanh could give reasonable accuracy.

(c)In the formula $M = AS$, the A and the S are both unknown, so the formula could be rewritten as $m = (\frac{1}{\alpha_i}) A_i (\alpha_i S_i)$, where α is any nonzero constant, the $\alpha_i S_i$ is a solution. This is the ambigiosity of the ICA. So a function named the Flip signal sign is provided in this module, which adjusts every element of the A to be non-negative values.

(d) When $d_m > d_s$, there is redundancy in the signal: the eigenvalue of the Covariance Matrix shows the connection of the original signals. When the eigenvalue is zero, there is linear dependency in the mixing signal and the ICA will remove that signal. The user could specify a threshold to remove a signal. Because the eigenvalue of the noise is much smaller than that of other signals, the ICA could be used for removing noise.

Properties

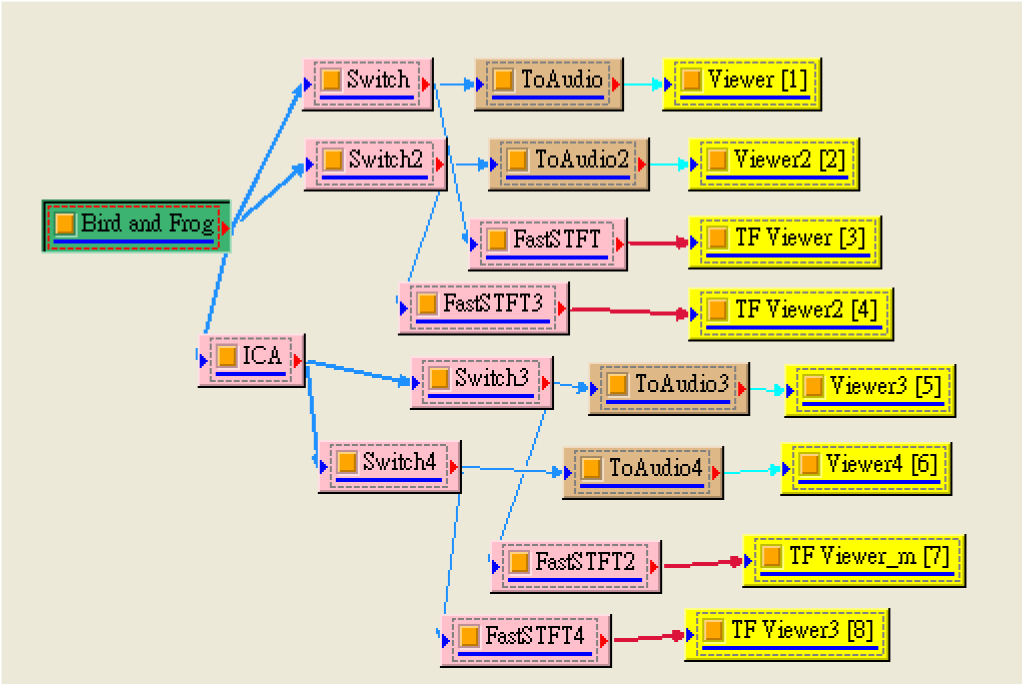
This module accepts real numbers, multi-channels, regular signals and audio signals.

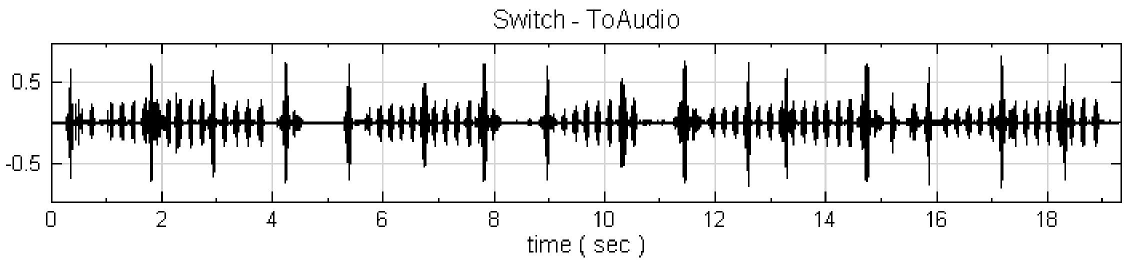
Property	
ICA Parameters	
ICA Method	**Symmetric**
Cost Function	**hyperbolicTan**
Max Iteration Steps	**100**
Epsilon	**0.0001**
Neglect Small Eigenvalue	**1E-08**
Flip Signal Sign	**True**
Number of Independent Components	0
Computed Iteration Steps	0
ICA Property	
Report...	**ICA Property Report**

Property Name	Property Definition	Default Value
ICA method	Setting the method of the ICA, Symmetric or Delfation.	Symmetric
Cost function	Setting the distribution function of the ICA hyperbolicTan, skewing or kurtosis.	hyperbolicTan
Max iteration Steps	Setting the maxmum time of the interation.	100
Epsilon	Setting the criterion for judgement of the convergence.	0.0001
Neglect small Eigenvalue	Setting the eigenvalue threshold for the redundant signal.	1E-08
Flip Signal Sign	Setting whether the sign inversion function is turnd on or not.	True
Number of Independent Components	Setting the number of Independent components.	User defined
ComputedIteration Steps	Setting the interation time for the computation.	User defined
Report	Setting whether the eigenvalue and the eigenvector are output in a report or not.	User defined

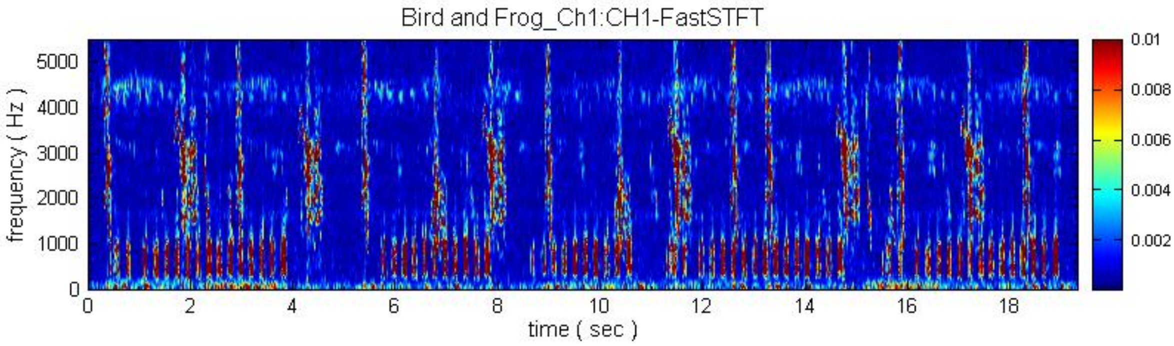
Example

Opening the demo79 (C:\Program Files \ DynaDx \ DataDemon \ demo \ Enhanced \ demo79 - ICA.vsn), the user could find the original signal named "Bird and Frog". It is a recording of two mics for the sound of two different frogs. One of the sounds is similar to the chiriping sound of the bird. The user could hear these two sounds by pressing the Play button on the up left corner of the Viewer.

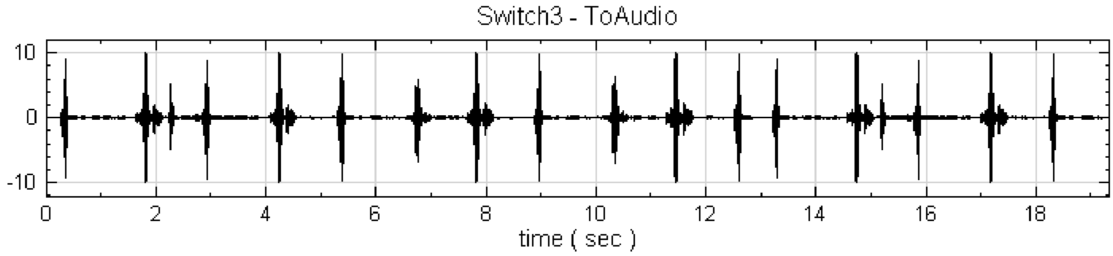


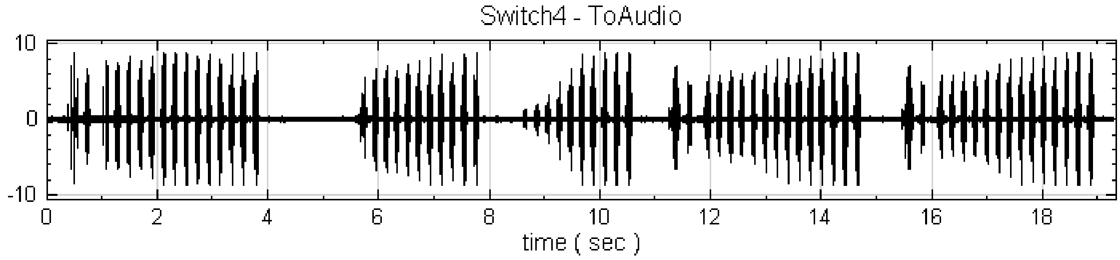


One sound is computed by the Fast STFT; the result is shown as below:



And then, decomposing the mixing signal by the ICA, the result is shown below:





References

1. Independent Component Analysis by by Aapo Hyvtirinen, Juha Karhunen, and Erkki Oja A Wiley-Interscience Publication

2. E. Bingham and A. Hyv¨arine,. A fast fixed-point algorithm for independent component analysis of complex-valued signals. Int. J. of Neural Systems, 10(1):1–8, 2000.

3. A. Hyv¨arinen. A family of fixed-point algorithms for independent component analysis.In Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97), pages 3917–3920, Munich, Germany, 1997.

4. A. Hyv¨arinen. Fast and robust fixed-point algorithms for inde7endent component analysis. IEEE Trans. on Neural Networks, 10(3):626–634, 1999.

5. Z. Koldovský, P. Tichavský and E. Oja, "Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining the Cramér-Rao Lower Bound", IEEE Trans. on Neural Networks, Vol. 17, No. 5, Sept 2006.

3.9 Matrix (Professional Only)

3.9.1 Matrix Operatoin

This module does calculations between two maxtrices A and B.

Matrix Addition: $A+B=a_{ij}+b_{ij}$, A and B have same dimensions M * N.

Matrix Subtraction: $A-B=a_{ij}-b_{ij}$, A and B have same dimensions M * N.

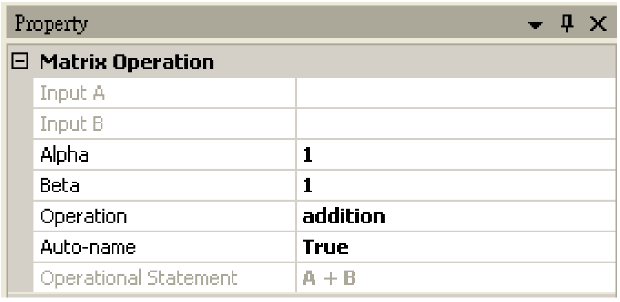
Matrix Multiplication: $A\cdot B=\sum_{k}a_{ik}\cdot b_{kj}$, A’s dimension is M * N and B’s dimension is N * P.

Matrix Left Division: $A\backslash B=A^{-1}B$, A’s dimension is M * N and B’s dimension is M * P.

Matrix Right Division: $A/B=AB^{-1}$, A’s dimension is M * N and B’s dimension is P * N.

Properties

This module accepts real number, complex number, and Numeric data. And the output has the same format as the input.



Property Name	Property Definition	Default Value
Input A	A's name	Reference only
Input B	B's name	Reference only
alpha	A's weight factor α, detailed below.	1
Beta	B's weight factor β, detailed below.	a
Operation	Operator: addition, subtraction, multiplication, left division, and right division.	addition
Auto-name	If true, the module name is A+B for A+B and it depends on the operator; Otherwise, it is Mop.	true
By Element	If true, calculation is for each element and not for the matrix.	False
Operation statement	Operation expression	Reference only

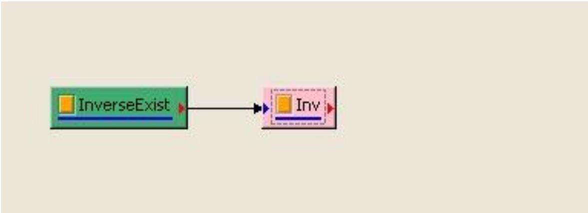
For matrix addition, the output matrix is C, $C = \alpha A + \beta B$, α and β are the weight factors for matrices A, B respectively.

Example

Use DoMatlab to create a 3 * 3 random matrix, the content is:

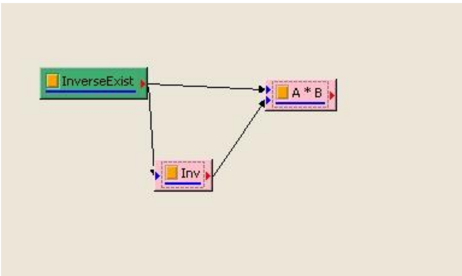
$$\begin{pmatrix} 0.86\ 0.63\ 0.37 \\ 0.22\ 0.66\ 0.69 \\ 0.99\ 0.56\ 0.78 \end{pmatrix}$$

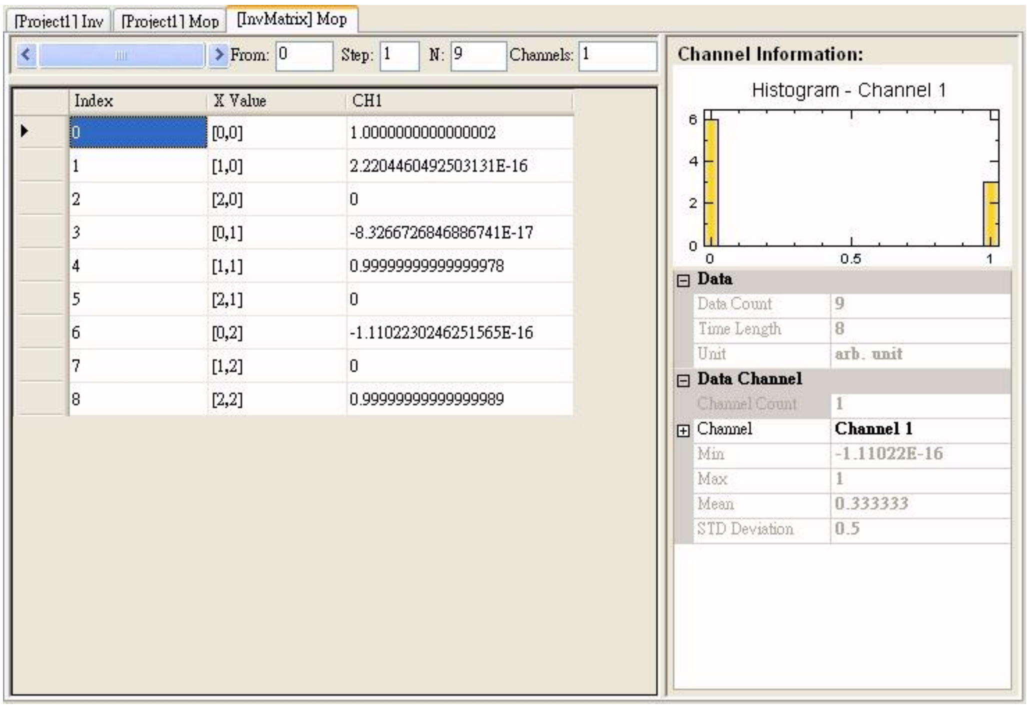
Get the Inverse Matrix and the result is:



$$\begin{pmatrix} -3.44 & 8.05 & -5.49 \\ 3.09 & -0.87 & -0.69 \\ -1.79 & 0.39 & 2.47 \end{pmatrix}$$

Do the multiplication for these two matrices, and a unit matrix is obtained.





Related Functions

Matrix Operation

References

3.9.2 Matrix Inverse

Calculate inverse matrix A^{-1} of A (A must be a square matrix), so $A \cdot A^{-1} = I$, where I is the unit matrix.

Properties

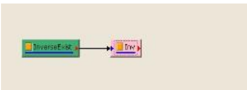
This module accepts real number, complex number, and Numeric data. And the output has the same format as the input.

Example

Use DoMatlab to create a 3 * 3 random matrix, the content is:

$$\begin{pmatrix} 0.86 & 0.63 & 0.37 \\ 0.22 & 0.66 & 0.69 \\ 0.99 & 0.56 & 0.78 \end{pmatrix}$$

Get the Inverse Matrix and the result is:



$$\begin{pmatrix} -3.44 & 8.05 & -5.49 \\ 3.09 & -0.87 & -0.69 \\ -1.79 & 0.39 & 2.47 \end{pmatrix}$$

Do the multiplication for these two matrices, and a unit matrix is obtained.

If the input matrix is Singular, such as $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, we get Warning message below,

Related Functions

Matrix Operation

3.9.3 Transpose

Calculate the transpose and conjugate (optiontional) of matrix A.

Properties

This module accepts real number, complex number, and Numeric data. And the output has the same format as the input.

Property Name	Property Definition	Default Value
Complex Conjugate	For complex element matrix, take conjugate calculation.	true

Example

The matrix A is 2 X 3 and its elements are:

$$\begin{pmatrix} 2+8i & 3 & 4+9i \\ 5 & 6 & 7 \end{pmatrix}$$

Calculate transpose without conjugate, the output matrix A^T is

$$\begin{pmatrix} 2+8i & 5 \\ 3 & 6 \\ 4+9i & 7 \end{pmatrix}$$

Calculate both transpose and conjugate, the output matrix A^+ is

$$\begin{pmatrix} 2-8i & 5 \\ 3 & 6 \\ 4-9i & 7 \end{pmatrix}$$

3.9.4 Extract Region of Interest

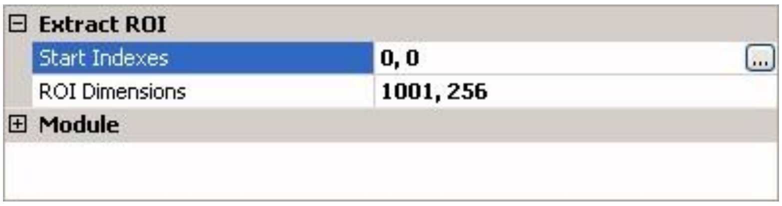
Extract a matrix S from a matrix A. The Start Indexes is [sx , sy], the row length is lx (Row) and column length is ly (Column) matrix, and the End Indexes is [sx + lx - 1, sy + ly - 1]. If the End Index exceeds the dimension of matrix A, 0 is filled.

Introduction

It is very close to the definition of Sub-Matrix. However, this module can handle with higher dimension matrix. The elements are filled with 0 if the dimenstion is larger than the original matrix.

Properties

This module accepts real number, complex number, and Numeric data. And the output has the same format as the input.



Property Name	Property Definition	Default Value
Start Indexes	Starting element for extraction	(0, 0)
ROI Dimensions	Row and Column length for extraction	Row and Column length of the original matrix

Example

Use DoMatlab to create a 3 * 3 random matrix, the elements are:

$$\begin{pmatrix} 0.86 & 0.63 & 0.37 \\ 0.22 & 0.66 & 0.69 \\ 0.99 & 0.56 & 0.78 \end{pmatrix}$$

Extract a matrix from starting element (0, 0) and ROI Dimensions (3, 1), the result is,

$$\begin{pmatrix} 0.78 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

If the starting element is (2, 2) and ROI Dimensions is (3, 3) (exceeds the dimension of the original matrix), the result matrix is

$$\begin{pmatrix} 0.78 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The elements outside the dimension of the original matrix are set to 0.

References

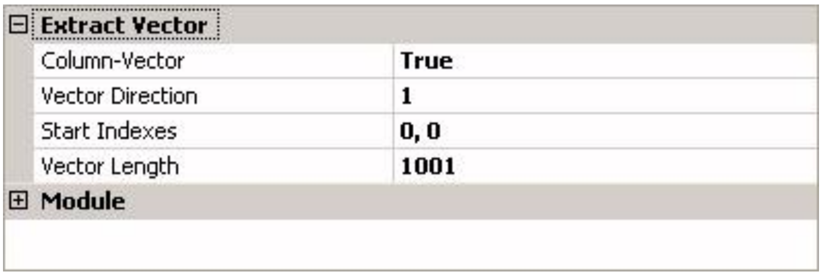
Gilbert Strang: Linear Algebra and Its Applications 3rd edition.

3.9.5 Extract Vector

Extract vector from matrix (not limit to two dimensions). If the extracted vector is larger than the matrix, elements are filled with 0 value.

Properties

This module accepts real number, complex number, and Numeric data. And the output has the same format as the input.



Property Name	Property Definition	Default Value
Vector Direction	The direction to extract vector from 0 position. If there is a 4-dimension matrix, the direction of 2 means to extract vector along the 3rd dimension (so called the depth).	0 (row direction)
Column-Vector	If true, the output vector is column based. Otherwise, the output is row based.	True
Start Indexes	The starting element for the extraction.	(0 , 0)
Vector Length	The length of the extraction	Length of the matix along a direction

Example

There is a 2 * 3 matrix A, its elements are:

$$\begin{pmatrix} 0.27 & 0.95 & 0.15 \\ 0.54 & 0.96 & 0.97 \end{pmatrix}$$

Set Column Vector to True, set Vector Direction to 0 (row direction), the starting point is (0,0), and the length of extraction is 3. The resulting vector is:

$$\begin{pmatrix} 0.27 \\ 0.54 \\ 0 \end{pmatrix}$$

References

Gilbert Strang: Linear Algebra and Its Applications 3rd edition.

3.9.6 Diagonal Vector

This module extracts diagonal elements from a square matrix and outputs as a vector.

Properties

This module accepts real number, complex number, and Numeric data. And the output has the same format as the input.

Property Name	Property Definition	Default Value
Column-Vector	If true, the output vector is column based. Otherwise, the output is row based.	True

Example

Use DoMatlab to create a 3 * 3 random matrix, the elements are:

$$\begin{pmatrix} 0.03 & 0.67 & 0.39 \\ 0.84 & 0.75 & 0.65 \\ 0.93 & 0.74 & 0.17 \end{pmatrix}$$

Extract Diagonal Vector and get a vector of

$$\begin{pmatrix} 0.03 \\ 0.75 \\ 0.17 \end{pmatrix}$$

References

Gilbert Strang: Linear Algebra and Its Applications 3rd edition.

3.9.7 Reciprocal Matrix Condition Number

Calculate 1 / C, where C is the Condition Number of the matrix.

Introduction

For a given matrix A, the Condition Number of matrix A is defined as $c=\|A\|\|A^{-1}\|$. C can be measured with L_1 or L_∞, which are called as C_1 and C_∞ respectively. They are corresponding to the Norm of A using different measurement. The calculation of $\|A\|_\infty$ and $\|A\|_1$ are defined below (which is different from Linear Algebra textbook):

$$\|A\|_\infty \triangleq \max_i \sum_j |a_{ij}| \quad , \quad \|A\|_1 \triangleq \max_j \sum_i |a_{ij}|$$

where i, j are the index of the row and column.

The meaning fo the Condition Number:

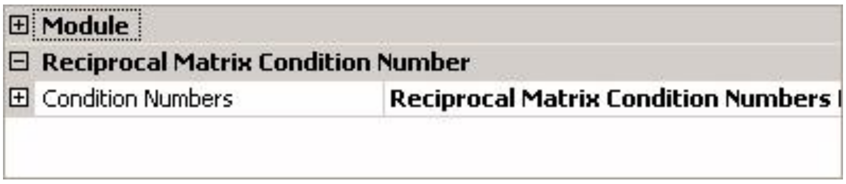
To measure the Stiffness of matrix A. This is the distribution of the eigen-value $\frac{\lambda_{max}}{\lambda_{min}}$.

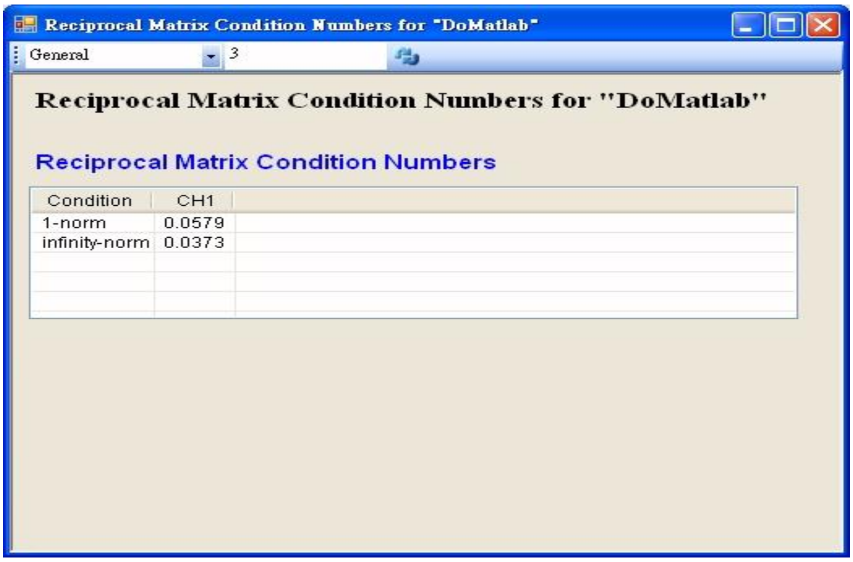
The round-off error sensitivity of the solution for the linear quation $Ax=b$, i.e. when b varies a little bit, how x changes, $A(x+\delta x)=b+\delta b$.

Note: A must be a square matrix. If not, SVD method can be applied. However, it is not supported in this version.

Properties

This module accepts real number, complex number, Numeric data. And the output is in the Reporter of the Properties.



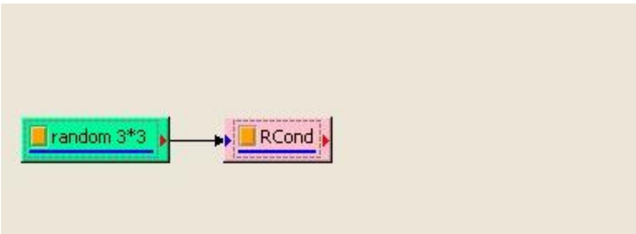


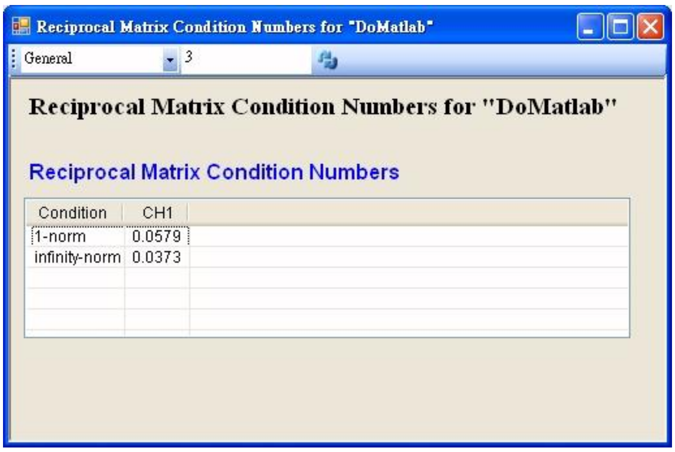
Example

Use DoMatlab module to createa matrix A with random elements:

$$\begin{pmatrix} 0.86 & 0.63 & 0.37 \\ 0.22 & 0.66 & 0.69 \\ 0.99 & 0.56 & 0.78 \end{pmatrix}$$

If its Condition Number is close to 0, this matrix is Singular. Its inverse matrix does not exist. We can test it by





It shows that the value of the Norm is not close to 0. So the inverse matrix does exist. It can be calculated via Inverse matrix module.

Related Functions

Matrix Inverse, Matrix Operation.

References

Gilbert Strang: Linear Algebra and Its Applications 3rd edition.

3.10 External (Professional Only)

3.10.1 External Dll

This module is to help users to call self developed algorithms and data input interface etc. Users can generate DLL (Dynamic Link Library) for their application using Visual C#, Visual Basic, and Visual C++ environments. The DataDemon can use this module to call these DLLs.

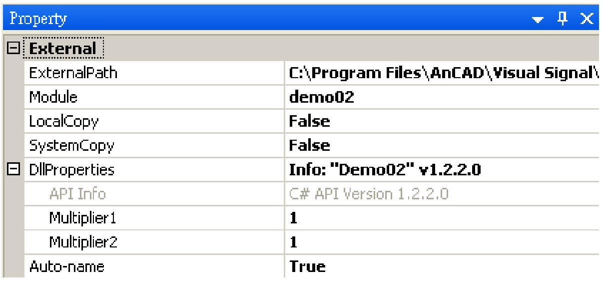
Introduction

Create DLL in Visual Studio environment first. Under Project, select “Class Library”, and add “vsmExternalBase.dll” to the References (C:\Program Files\DynaDx\DataDemon\External\vsmExternalBase.dll), remember "using VSignal.ExternalBase; "

The written Class must inherit TExternalBase, and modify two major methods Init( ) and DoCompute( ). The name, parameter, and modules etc of Init() method need to be set. The modification of DoCompute( ) is to accept input signal, use user’s algorithm for data processing, and output setting etc.

All settings can refer to ExternalBase Class Library section. The example of DLL can refer to C:\Program Files\DynaDx\DataDemon\External\api\cs\VSignalExternalDllDemo.

Properties

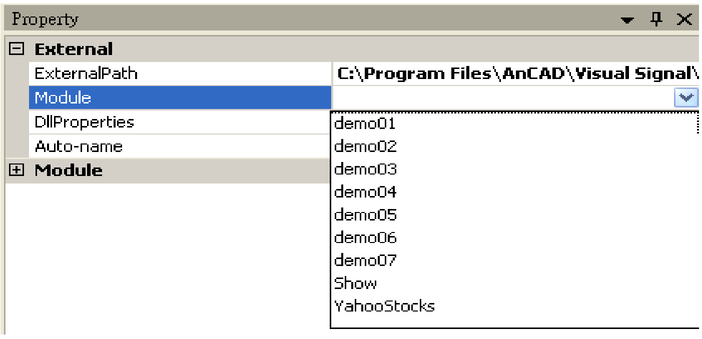


Property Name	Property Definition	Default Value
External Path	The directory to the external DLL	None
Module	Select module in the external DLL	None
LocalCopy	If “True”, copy DLL file to local project	False
SystemCopy	If “True”, copy DLL file to DataDemon special folder	False

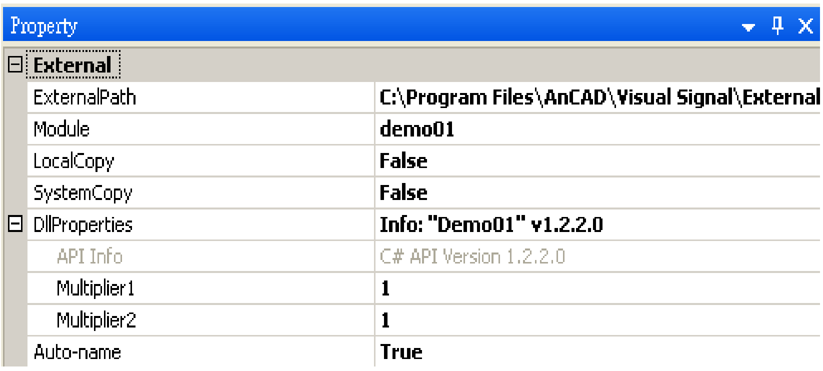
	(C:\Program Files\DynaDx\DataDemon\External\...)	
Auto-name	Set the name for the external modue automatically	True
DllProperties	Properties of DLL which implemented by the user, or API Version etc.	None

Example

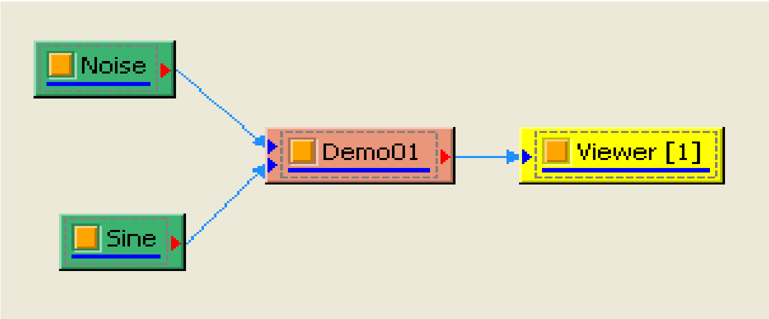
Start Compute / External / ExternalDll, and open ExternalDllDemos.dll in ExternalDLL path (C:\Program Files\DynaDx\DataDemon\External\External\ExternalDllDemos.dll). There are many modules in the Property / Module as shown below.

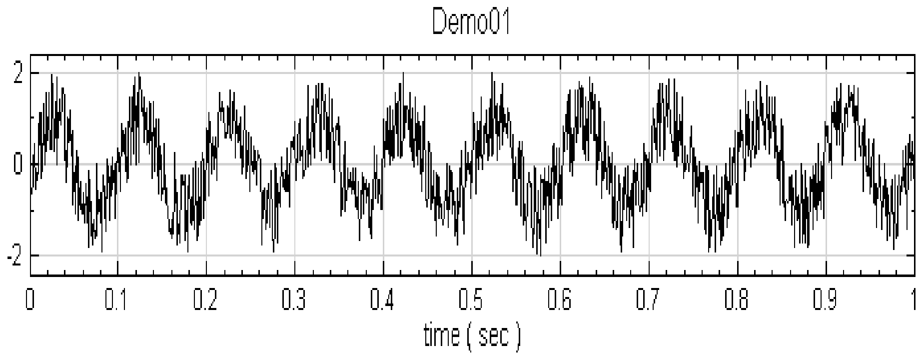


Select module demo1. In the DllProperties, Multiplier1 is the weight factor for the 1st signal and Multiplier2 is the weight factor for the 2nd signal. The sum of both weighted signals is the output.

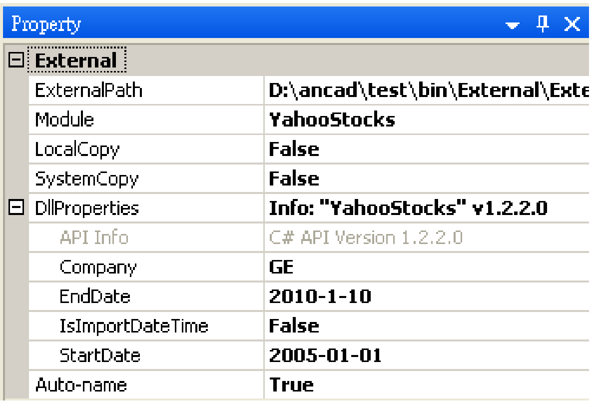


Connect both Noise and Sine Wave signal to demo1 SFO, then view the results using Channel Viewer.



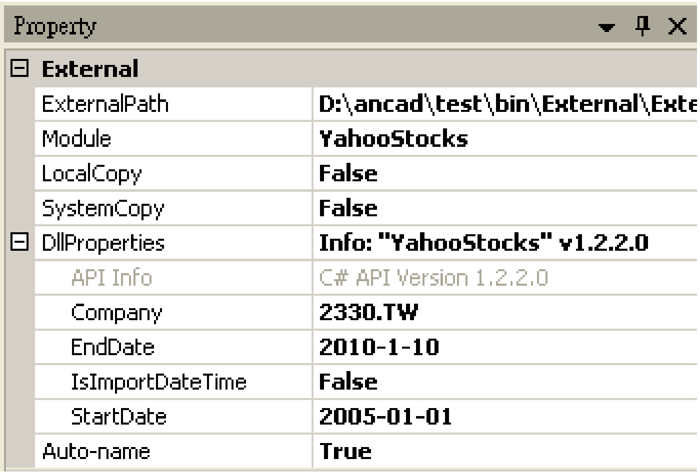


Open another ExternalDll. In ExternalPath directory, open ExternalDllDemos.dll (C:\Program Files\DynaDx\DataDemon\External\External\ExternalDllDemos.dll). And select YahooStocks in the Module. This DLL can download stock data from Yahoo website.


Property

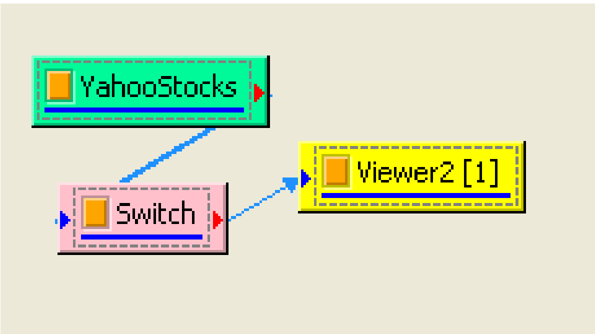
External	
ExternalPath	D:\ancad\test\bin\External\Exte
Module	YahooStocks
LocalCopy	False
SystemCopy	False
DllProperties	Info: "YahooStocks" v1.2.2.0
API Info	C# API Version 1.2.2.0
Company	GE
EndDate	2010-1-10
IsImportDateTime	False
StartDate	2005-01-01
Auto-name	True

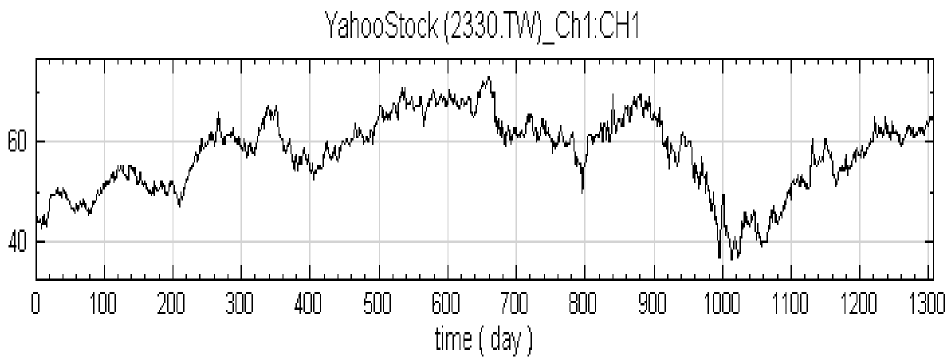
In DllProperties, set Company to 2330.TW. And connec it to Channel Switch and select open price, close price, volume etc, then display the data using Channel Viewer.



Property

External	
ExternalPath	D:\ancad\test\bin\External\Exte
Module	YahooStocks
LocalCopy	False
SystemCopy	False
DllProperties	Info: "YahooStocks" v1.2.2.0
API Info	C# API Version 1.2.2.0
Company	2330.TW
EndDate	2010-1-10
IsImportDateTime	False
StartDate	2005-01-01
Auto-name	True





Related Functions

ExternalViewer

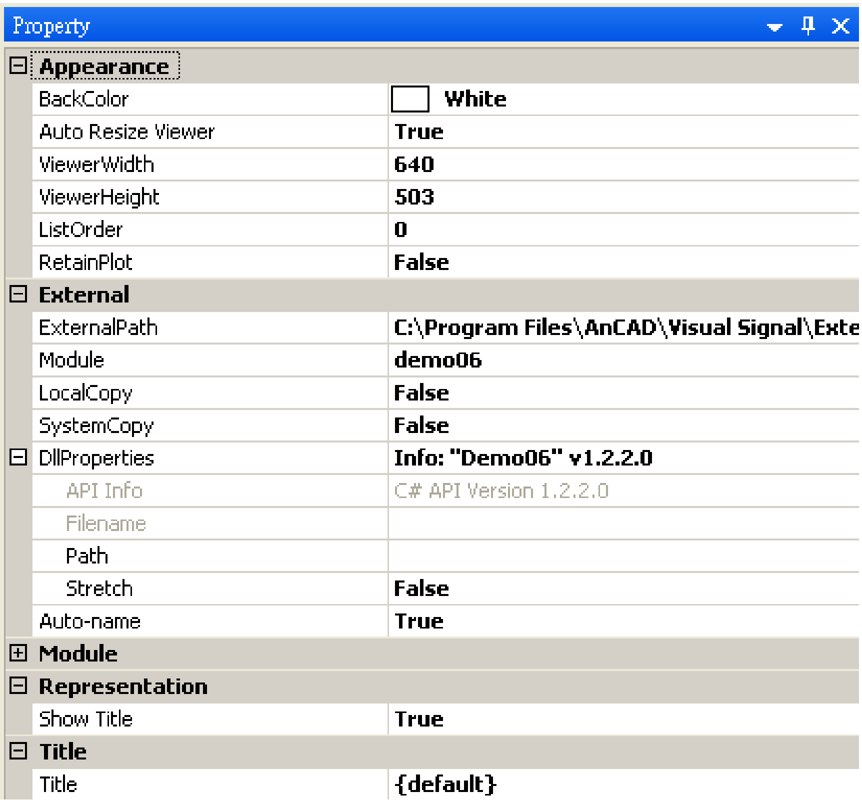
3.10.2 External Viewer

This module helps users to show different images in DataDemon. User can create different DLL (Dynamic Link Library) under Visual C#, Visual Basic, and Visual C++ environment. This module can display the images using these DLL.

Introduction

See the description in section 3.10.1.

Properties

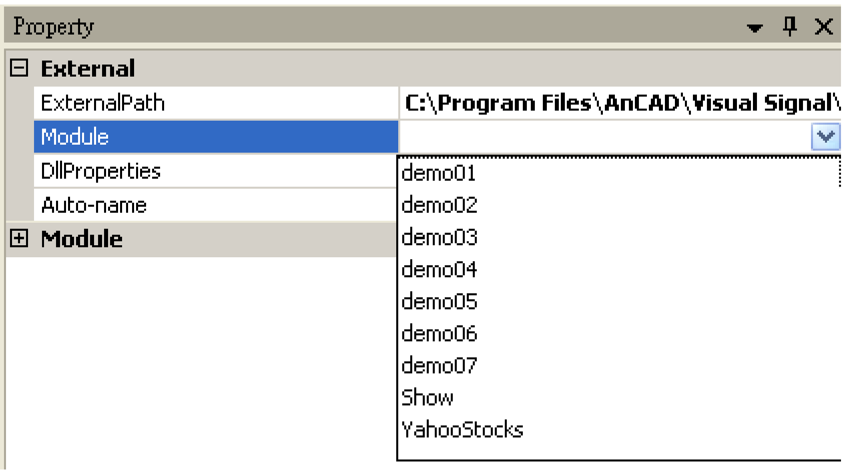


Property Name	Property Definition	Default Value
External Path	The directory to the external DLL	None
Module	Select module in the external DLL	None
LocalCopy	If “True”, copy DLL file to local project	False
SystemCopy	If “True”, copy DLL file to DataDemon special folder (C:\Program Files\DynaDx\DataDemon\External\...)	False

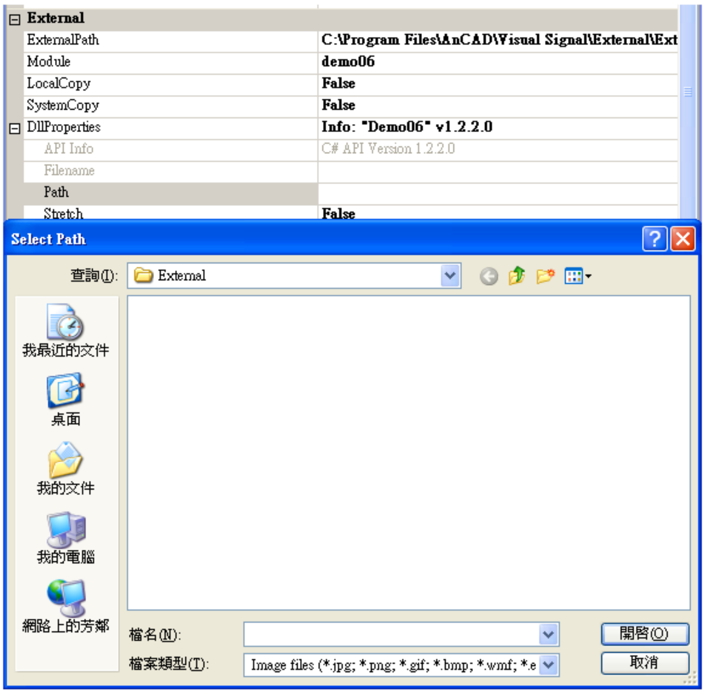
Auto-name	Set the name for the external modue automatically	True
DllProperties	Properties of DLL which implemented by the user, or API Version etc.	None
BackColor	Set background color of the plot	White
Auto Resize Viewer	If “True”, set the size of the plot based on the monitor display; Otherwise, set by the user.	False
ViewerWidth	Set the width of the plot, unit is in pixel.	default (750)
ViewerHeight	Set the height of the plot, unit is in pixel.	default (180)
ListOrder	Set the order to show the plots	Follow the order of Viewer creation
RetainPlot	Keep the plot. If True, the plot is kept even if the ChanneViewer is removed; if False, the plot is deleted.	False
Show Title	Show the title in the plot	True
Title	Title name	{default}

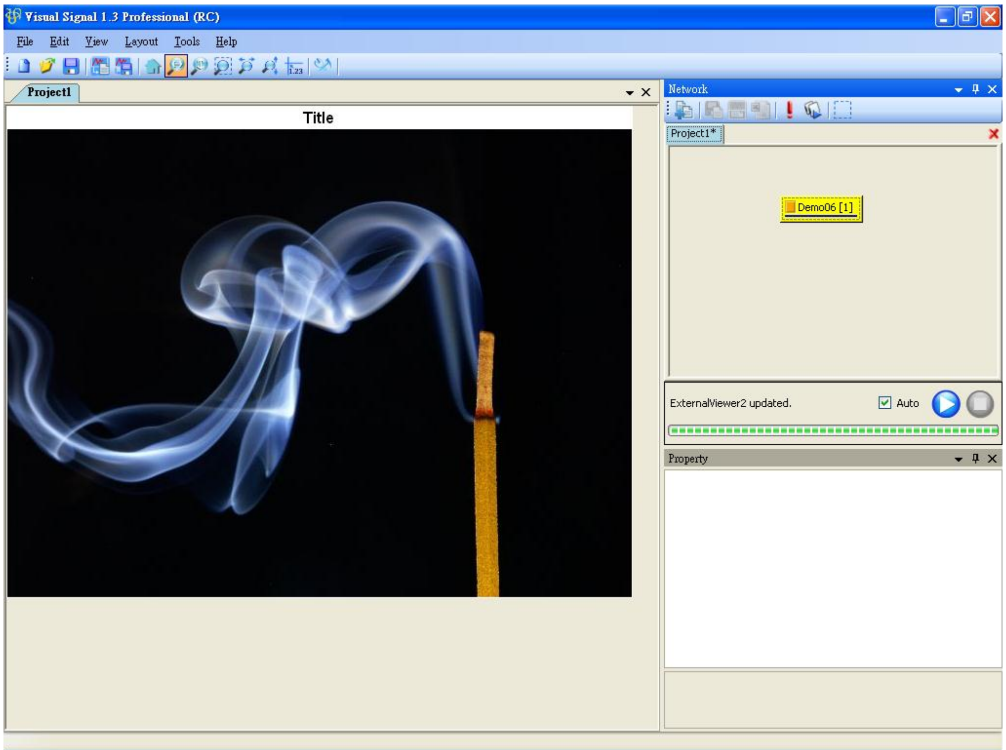
Example

Start Compute / External / ExternalDll, and open ExternalDllDemos.dll in ExternalDLL path (C:\Program Files\DynaDx\DataDemon\External\External\ExternalDllDemos.dll). There are many modules in the Property / Module as shown below.



Select demo6 in Module, Path property is shown in DllProperties. Select an image and display it using ExternalViewer.





Related Functions

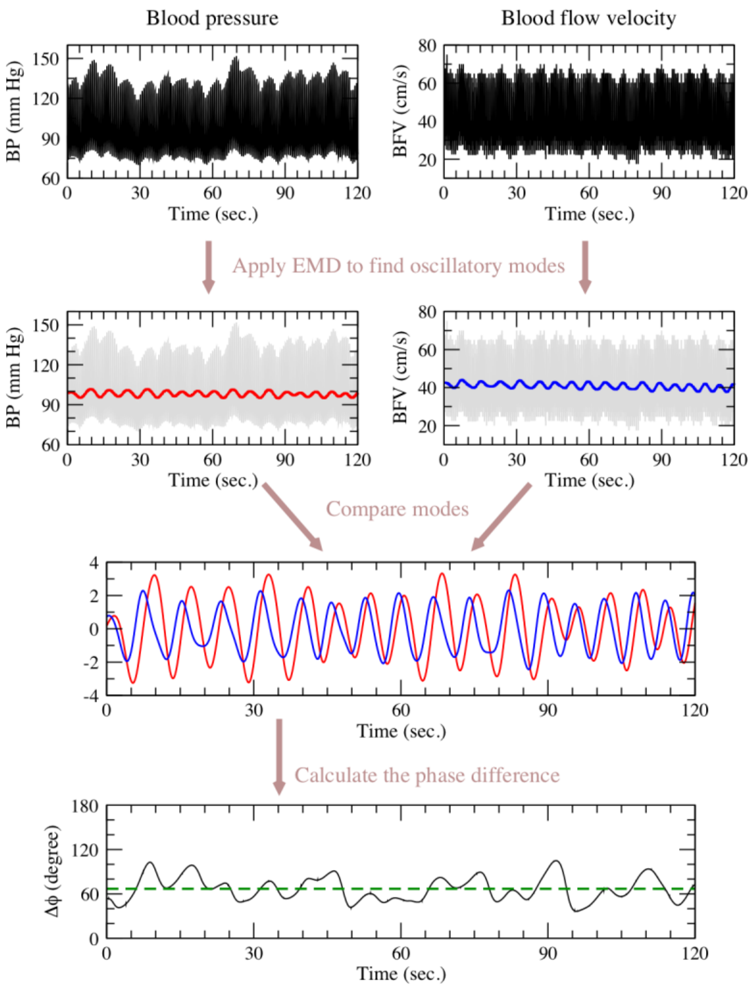
ExternalDll

3.11 MMPF (Multimodal Pressure-Flow)

Cerebral autoregulation reflects the ability of the cerebral microvasculature to adapt to systemic blood pressure (BP) changes by modulating the small vessel resistance to maintain relatively stable blood flow. Noninvasive assessment of cerebral vasoregulation is important for medical diagnostics and acute care. Recent studies have demonstrated that beat-to-beat measurements of BP and cerebral blood flow velocities (BFV) measured by transcranial Doppler ultrasound (TCD) during the Valsalva maneuver and head-up tilt can identify impairment of cerebral vasoreactivity in various medical conditions; indicating that a reliable, non-invasive index for dynamic cerebral autoregulation may be extracted from the BP and BFV signals.

Conventional approaches model autoregulation with BP as input and blood flow as output (e.g., Windkessel models) and assume that signals are composed of superimposed sinusoidal oscillations of constant amplitude and period over a pre-selected frequency range. A transfer function is typically used to explore the relationship between BP and BFV by calculating gain and phase shift between their spectra. However, BP and BFV signals are often nonstationary, and are modulated by nonlinearly interacting processes at multiple time-scales corresponding to the beat-to-beat systolic pressure, respiration, spontaneous BP fluctuations, and those induced by interventions such as the Valsalva maneuver and postural changes. To overcome problems related to nonstationarity and nonlinearity, Novak et al. [1] recently developed a novel computational method, the Multimodal Pressure-Flow (MMPF) technique to analyze the BP-BFV relationships during the Valsalva maneuver. Unlike conventional approaches that are based on the Fourier transform and thus require linearity and stationarity of the signals, the MMPF analysis does not rely on these assumptions.

Instead, the MMPF technique evaluates autoregulatory dynamics based on instantaneous phase analysis of BP and BFV oscillations. The MMPF analysis applies the Empirical Mode Decomposition (EMD) algorithm proposed by Huang et al. [2] to decompose complex BP and BFV signals into multiple empirical modes. Each mode represents a frequency-amplitude modulation in a narrow frequency band that can be related to a specific physiologic process. As a result, MMPF analysis not only can be applied to protocols such as the Valsalva maneuver and sit-to-stand conditions, where large BP and BFV oscillations were induced by the interventions, but can also be used to study spontaneous oscillations of BP and BFV under resting ("baseline") conditions [3].



The above figure is an illustration of the MMPF procedure. Continuous BP and BFV signals (panels in the 1st row) of a healthy subject collected under baseline supine conditions are used for this example. The dominant oscillations of BP and BPV, due to physiologic breathing can be extracted by the EMD algorithm (panels in the 2nd row). These two oscillatory modes can then be compared with each other as shown in the 3rd row panel. Note that the oscillatory component of the BFV (blue curve) consistently leads that of the BP (red curve). This phase relationship is an important marker of healthy autoregulation. The instantaneous phases of these two oscillations can be calculated by the Hilbert transform, and their difference is shown in the bottom panel. As apparent in this 2-minutes period, the phase shift between these two oscillations varies around an average value of 67 degrees (indicated by a dark green dashed line). Note that the example shown here reveals a relatively slow respiratory period (cycle length ~ 7 sec.). Typical subjects have a faster breathing rate; however, similar BFV/BP phase shift behavior is observed. In contrast, pathologic impairments of dynamic cerebral autoregulation can significantly reduce this phase shift. Therefore, the phase shift index may serve as a sensitive biomarker of autoregulation.

Obtaining a reliable index of dynamic cerebral autoregulation is useful under many clinical conditions. Impairment of vascular reactivity to regulate cerebral perfusion has been found in many syndromes associated with aging, hypertension, stroke, diabetes, dementia and traumatic brain injury (TBI). For example, autoregulation

estimates based on BFV and direct measurements of cerebral perfusion pressure (CPP) have shown predictive value for determining outcomes in TBI patients. Therefore, the MMPF analysis may play an important role to assess and monitor dynamic cerebral autoregulation in a wide range of clinical settings [1,3,4].

References

1. Novak V, Yang ACC, Lepicovsky L, Goldberger AL, Lipsitz LA, Peng C-K. Multimodal pressure-flow method to assess dynamics of cerebral autoregulation in stroke and hypertension. Biomed. Eng. Online 2004;3:39.

2. Huang NE, Shen Z, Long SR, et al. The empirical mode decomposition method and the Hilbert spectrum for non-stationary time series analysis. Proc. Roy. Soc. London 1998;A454:903-995.

3. Hu K, Peng C-K, Huang NE, Wu Z, Lipsitz LA, Cavallerano J, Novak V. Altered phase interactions between spontaneous blood pressure and flow fluctuations in type 2 diabetes mellitus: Nonlinear assessment of cerebral autoregulation. Physica A 2008;387:2279-2292.

4. Hu K, Peng C-K, Czosnyka M, Zhao P, Novak V. Nonlinear assessment of cerebral autoregulation from spontaneous blood pressure and cerebral perfusion pressure fluctuations. Cardiovasc. Eng. 2008;8:60-71.

There are several functions associated with MMPF module：

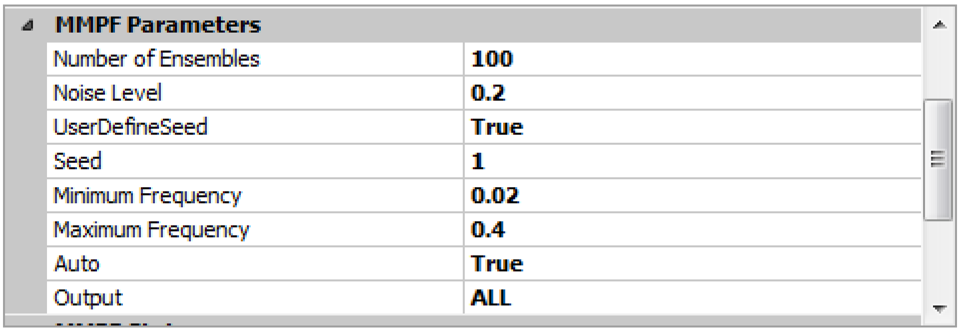
1. MMPF：It's different from the method above.It Estimates mean phase differences over each cycle for BP and BFV.
2. MMPF Imf：Used to extract IMFs from MMPF output.
3. MMPF PhaseDiff：Used to extract phase differences from MMPF output.
4. ImfPhaseDiff：A method to compute phase differences on two single-channel IMF signals.
5. MMPF Auto Macro：A user-defined macro.
6. MMPF Expert Macro：A user-defined macro.
7. TFA (Transfer Function Analysis)：The method was introduced for the analysis of continuous recordings of BP/BFV and is widely used.
8. TFAPhaseDiff： Used to extract phase differences from TFA output.

3.11.1 MMPF

The algorithm contains 2 steps: 1st, use RCADA EEMD to calculate IMFs from BP and BFV singals; 2nd, calculate mean phase differences over each cycle for BP and BFV.

Properties

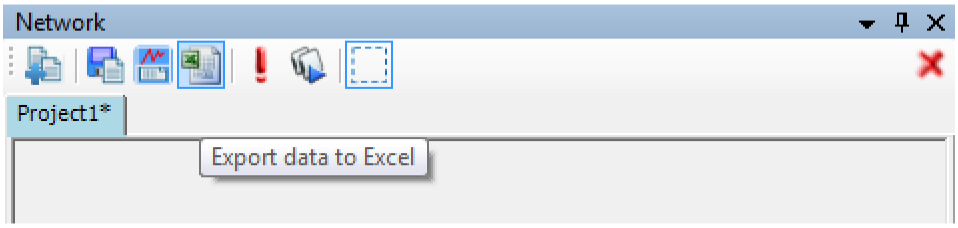
This module accepts input signal of real number, single channel, Regular and Audio.The proprertise are introduced below.



Property Name	Property Definition	Default Value
Number of Ensembles	The number of ensembles	100
Noise Level	The amplitude level of the normally distributed noise added to the original signal for ensemble computation	0.02
UserDefineSeed	Specify the control of noise seed, If false, the seed would be set according to current time.	True
Seed	Specify the initial value of the noise seed. Hidden, when "UserDefineSeed" is False.	1
BP Selected Mode	Selected mode for BP. Hidden, when "Auto" is True.	1
BFV #1 Selected Mode	Selected mode for BFV 1. Hidden, when "Auto" is True.	1
BFV #2 Selected Mode	Selected mode for BFV 2. Hidden, when "Auto" is True.	1
Minimum Frequency	Minimum frequency of the dominant oscillation range in Hz.	0.02

Maximum Frequency	Maximum frequency of the dominant oscillation range in Hz.	0.4
Auto	Specify "True" for automatic IMF mode selection.	True
Output	Select output for MMPF results, IMF or both of them.	ALL

You can click the ‘Excel’ button to export data of all results or selected results to a file for futher analysis.



The tables below explain the meaning of each column in the output Excel file:

Define: NBP = total number of IMF modes of BP (including residual)

NBV1 = total number of IMF modes of **left** BFV (including residual)

NBV2 = total number of IMF modes of **right** BFV (including residual)

If there is no BFV2 input for MMPF analysis, NBV2 = 0; and there will be no columns related to the right BFV in output file.

NT= NBP + NBV1+ NBV2

1. Select output for IMF

Column	Description
CH1~CH(NBP)	each mode for BP (Mode 1 is the raw input data)
CH(NBP+1)~CH(NBP+NBV1)	each mode for BFV1 (Mode 1 is the raw input data)
CH(NBP+NBV1+1)~CH(NT)	each mode for BFV2 (Mode 1 is the raw input data) (If BFV2 input exists)

CH(NT+1)	Row[1]= NBP Row[2]= NBV1 Row[3]= NBV2(If there is no BFV2 input, -1)

2. Select output for MMPF

Column	Variables	Description
CH1	IMFmodeBP	Chosen mode for BP
CH2	IMFmodeLBFV	Chosen mode for left BFV
CH3	IMFmodeRBFV	Chose mode for right BFV (If input BFV2 exists)
CH4	annoT_BP	To testify whether there is huge instantaneous frequency jump for BP (0: number of bad points >10)
CH5	annot_LBFV	To testify whether there is huge instantaneous frequency jump for LBFV (0: number of bad points >10)
CH6	annot_RBFV	To testify whether there is huge instantaneous frequency jump for RBFV (0: number of bad points > 10) (If input BFV2 exists)
CH7	cycleL(sec)	Period or length of the cycle
CH8	phaseshiftL	Estimated mean phase difference over each cycle for BP and LBFV
CH9	phaseshiftR	Estimated mean phase difference over each cycle for BP and RBFV (If input BFV2 exists)
CH10	errorL(pts)	The number of points of the phase difference berween BP and LBFV for consecutive two samples data points are over 0.8pi or under -0.8pi. The value should be equal to 0.
CH11	errorR(pts)	The number of points of the phase difference berween BP and RBFV for consecutive two samples data points are over 0.8pi or under -

		0.8pi. The value should be equal to 0. (If BFV2 input exists)
CH12	RatiocycleL	The ratio of mean frequency of BP to that of LBFV over each cycle.
CH13	RatiocycleR	The ratio of mean frequency of BP to that of RBFV over each cycle. (If input BFV2 exists)
CH14	ampRBP	The ratio of instantaneous amplitude over each cycle to STD of the corresponding IMF of BP
CH15	ampLBFV	The ratio of instantaneous amplitude over each cycle to STD of the corresponding IMF of LBFV.
CH16	ampRBFV	The ratio of instantaneous amplitude over each cycle to STD of the corresponding IMF for RBFV. (If input BFV2 exists)
CH17	ini_T	The initial time of identified cycle
CH18	end_T	The end time of identified cycle
CH19	ifnormalB	To test whether this cycle is under normal or arrhythmia epoch
CH20	ratio_Bad_point_BP	Ratio of number of bad points (huge frequency jump) to cycle length for BP
CH21	ratio_Bad_points_LBFV	Ratio of number of bad points (huge frequency jump) to cycle length for LBFV
CH22	ratio_Bad_points_RBFV	Ratio of number of bad points (huge frequency jump) to cycle length for RBFV. (If input BFV2 exists)
CH23	ratio_BP_LBFV	The ratio of number of points with phase difference between BP and LBFV >0.8pi or <-0.8pi to cycle length
CH24	ratio_BP_RBFV	The ratio of number of points with phase difference between BP and RBFV >0.8pi or <-0.8pi to cycle

		length. (If input BFV2 exists)
CH25		Row[5] = the number of cycles Row[6] = column ID of phase shift for BP-LBFV Row[7] = column ID of phase shift for BP-RBFV (If there is no input BFV2, -1)

3. Select output for ALL

Column	Description
CH1~CH(NT)	IMF results
CH(NT+1)~CH(NT+24)	MMPF results
CH(NT+25)	Row[1]= NBP Row[2]= NBV1 Row[3]= NBV2 (If there is no BFV2 for MMPF Input, -1) Row[4]= NT+1 (Identify the first column of MMPF results) Row[5]= the number of cycles Row[6]= column ID of phase shift for BP-LBFV Row[7]= column ID of phase shift for BP-RBFV (If there is no input BFV2, -1)

Related Functions

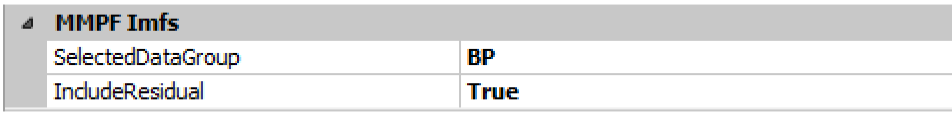
RCADA EEMD, Hilbert Transform

3.11.2 MMPFImf

MMPFImf is used to extract IMFs from MMPF output.

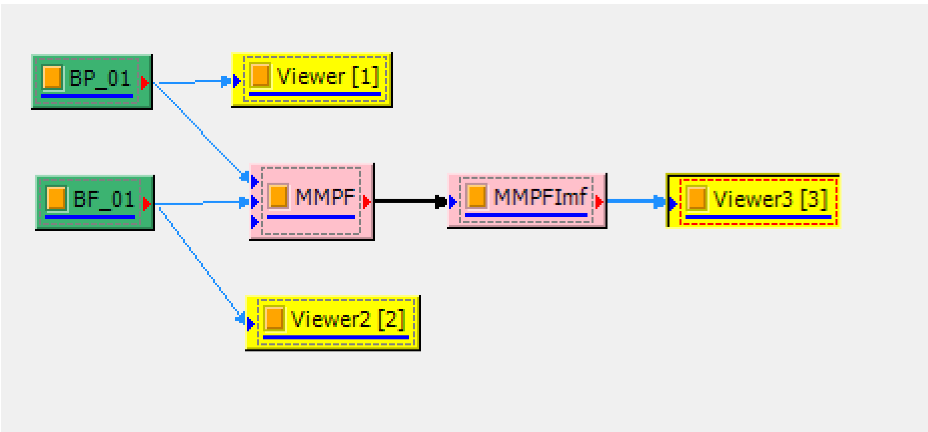
Properties

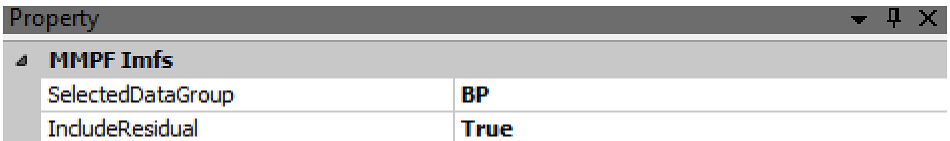
This module accepts input signal of real number, multiple channel, Regular. The properties are introduced below.



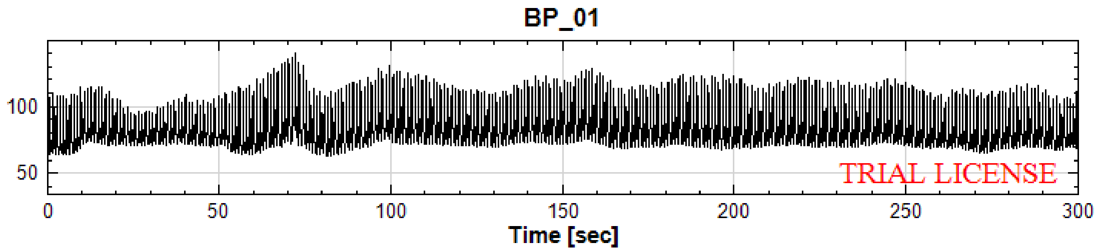
Property Name	Property Definition	Default Value
SelectedDataGroup	Select a data group (BP, BFV1 or BFV2).	BP
IncludeResidual	Specify "True" to include the residual IMF in the output	True

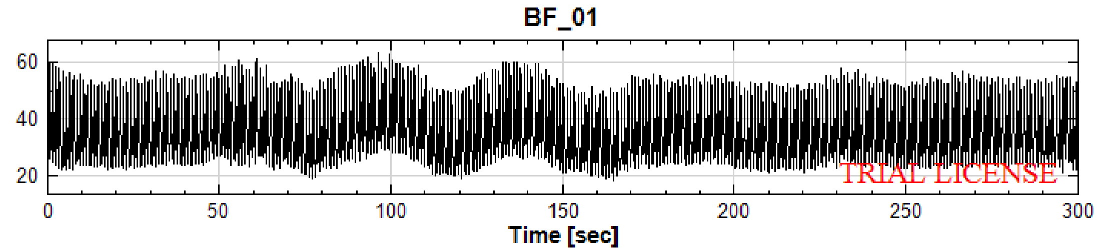
Example



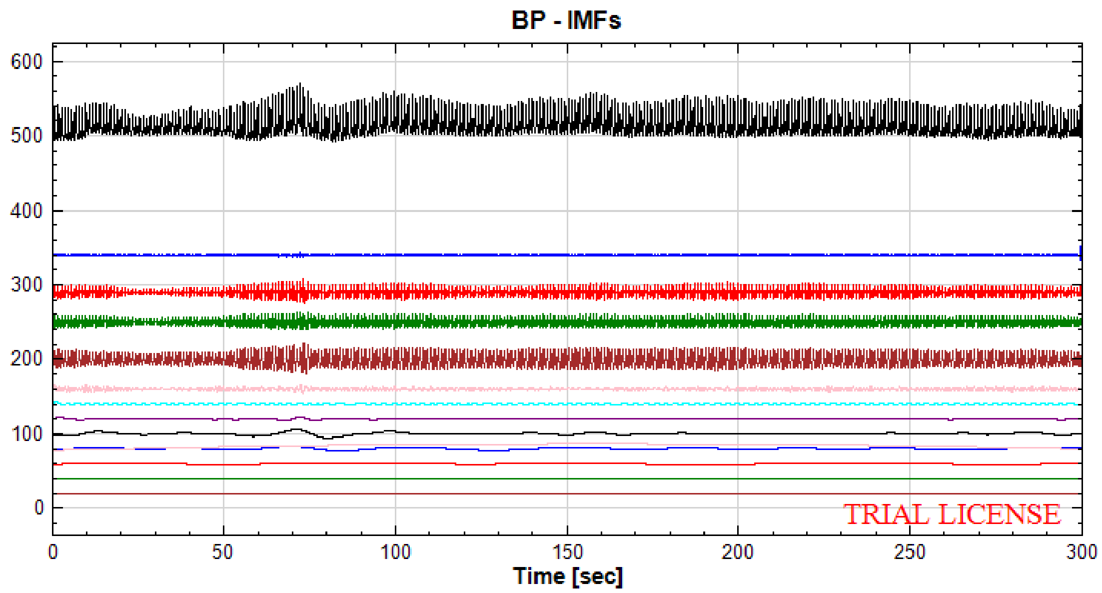


You can refer to demo95 in C:\ Program Files\ DynaDx\ DataDemon \ demo\ MMPF. The data used in this example is also in this folder.





The extracted BP IMFs is as following:



Related Functions

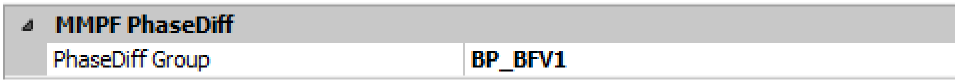
RCADA EEMD

3.11.3 MMPFPhaseDiff

MMPFPhaseDiff is used to extract phase differences from MMPF output.

Properties

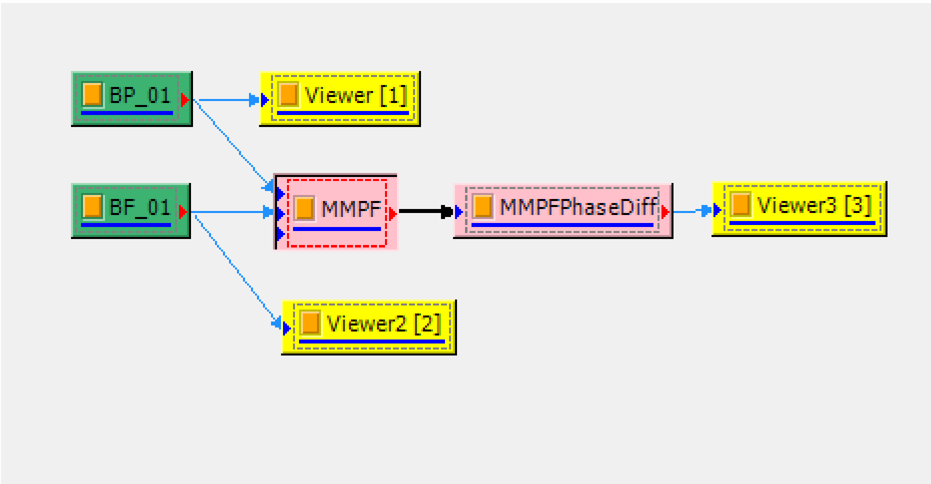
This module accepts input signal of real number, multiple channel, Regular. The properties are introduced below.

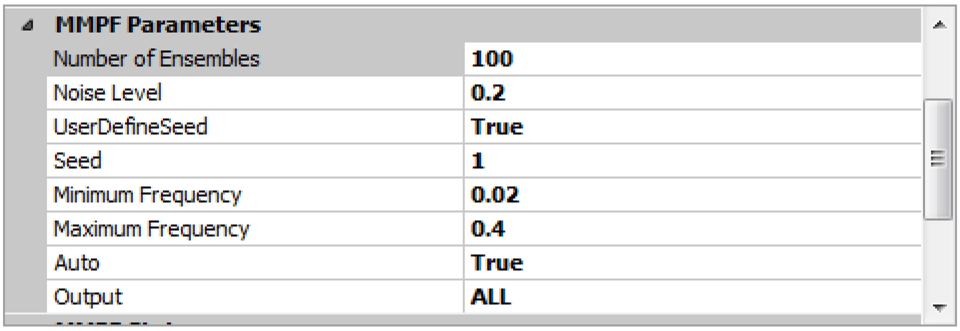


MMPF PhaseDiff	
PhaseDiff Group	BP_BFV1

Property Name	Property Definition	Default Value
PhaseDiff Group	Select a phase shift group(BP_BFV1 or BP_BFV2).	BP_BFV1

Example



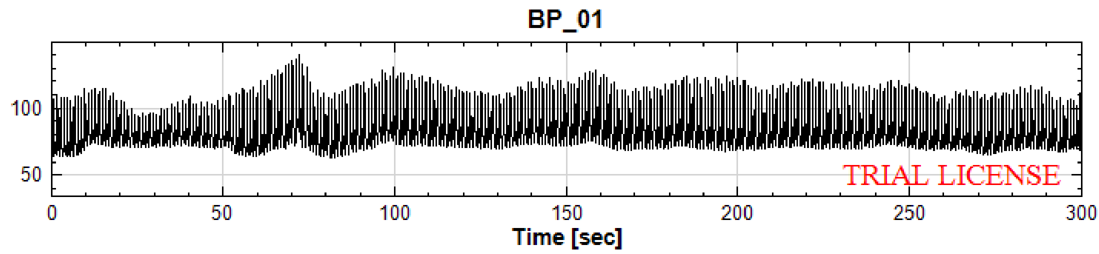


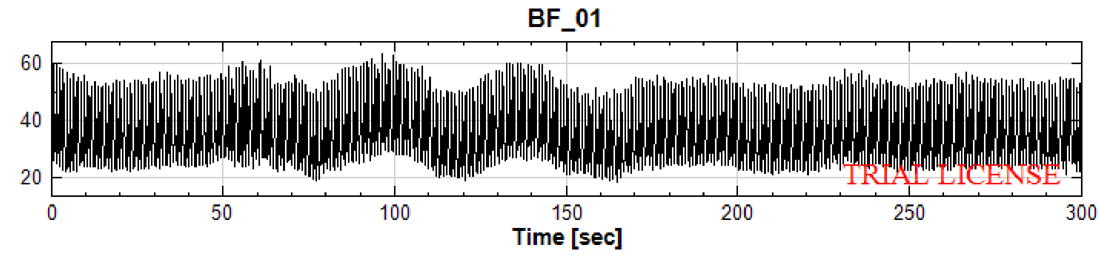
MMPF Parameters	
Number of Ensembles	100
Noise Level	0.2
UserDefineSeed	True
Seed	1
Minimum Frequency	0.02
Maximum Frequency	0.4
Auto	True
Output	ALL

MMPF PhaseDiff	
PhaseDiff Group	BP_BFV1

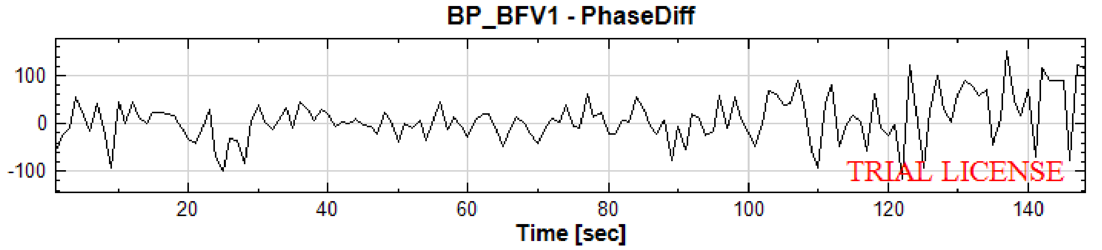
You can refer to demo96 in C:\ Program Files\ DynaDx\ DataDemon \ demo\ MMPF.

The data used in this example is also in this folder.





The extracted BP_BFV1 phase difference is as the following:



Related Functions

RCADA EEMD, Hilbert Transform

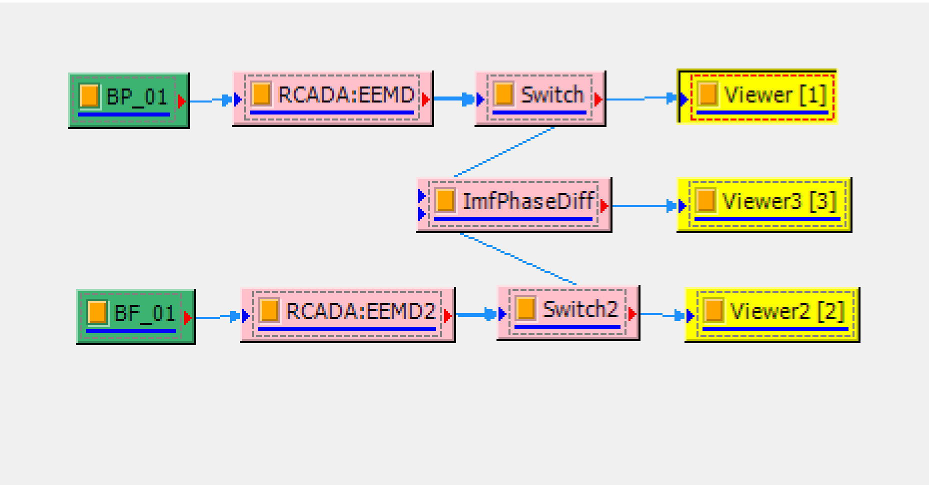
3.11.4 ImfPhaseDiff

ImfPhaseDiff is used to compute phase difference between two single-channel IMF signals.

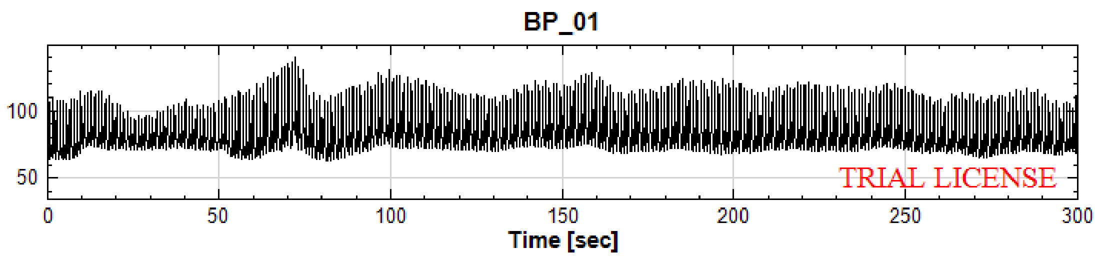
Properties

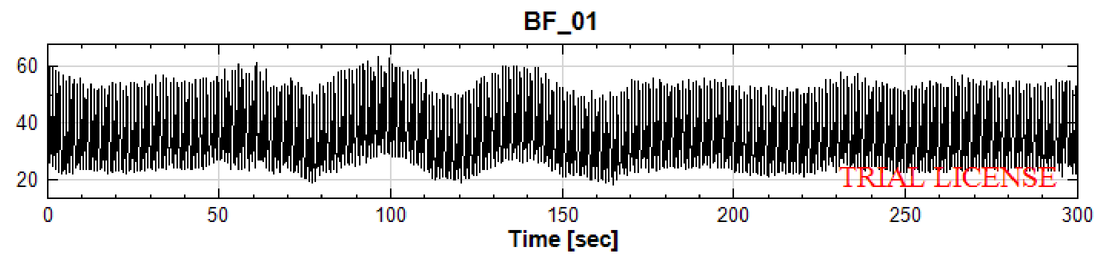
This module has no property.

Example

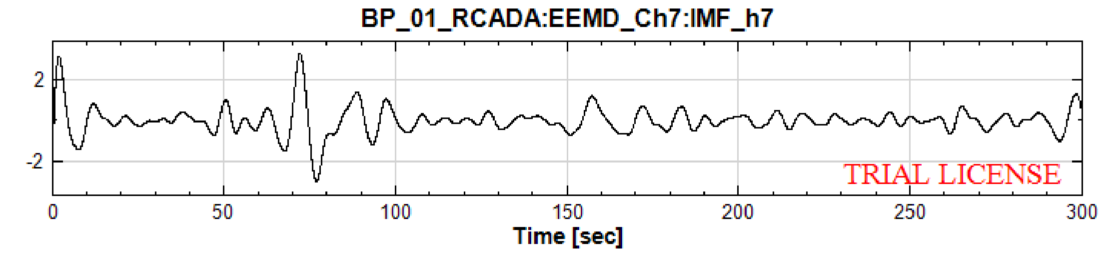


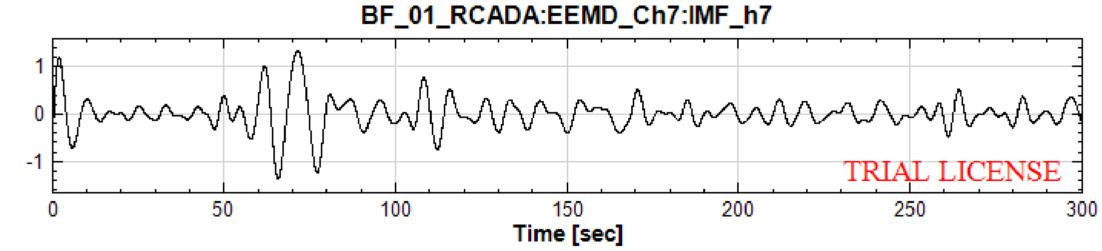
You can refer to demo97 in C: \ Program Files\ DynaDx\ DataDemon \ demo\ MMPF. The data used in this example is also in this folder.



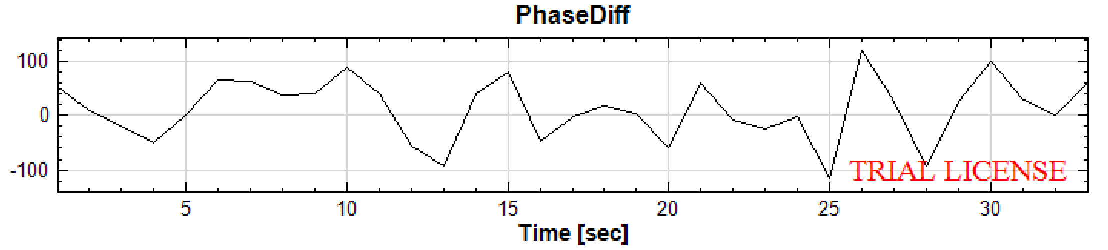


The selected IMF of BP and BFV is shown below:





Based on the selected IMFs from BP and BFV, the comuted phase difference is:



Related Functions

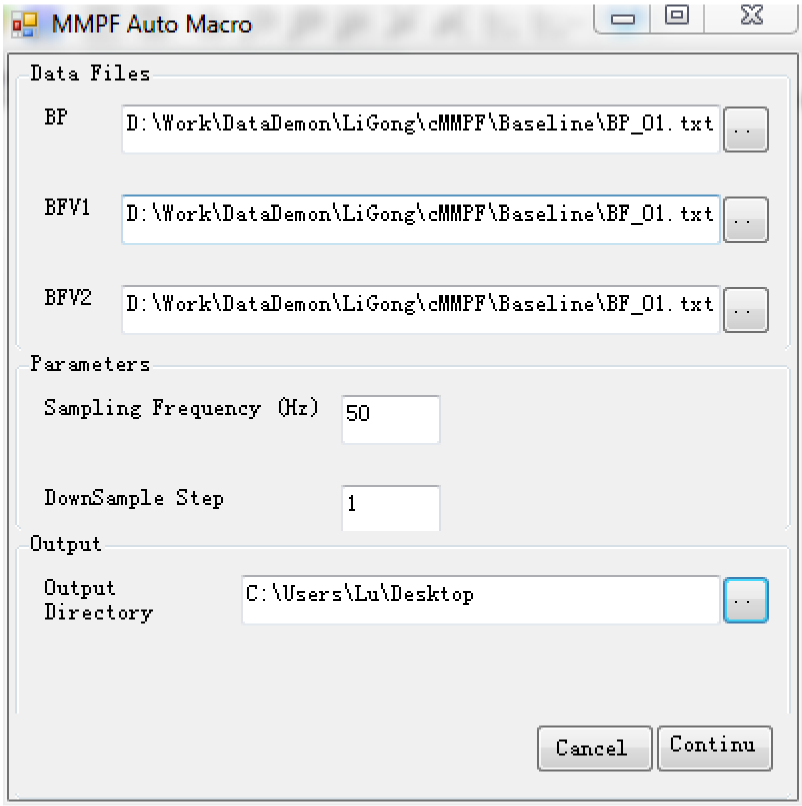
RCADA EEMD, Hilbert Transform

3.11.5 MMPF Auto Macro

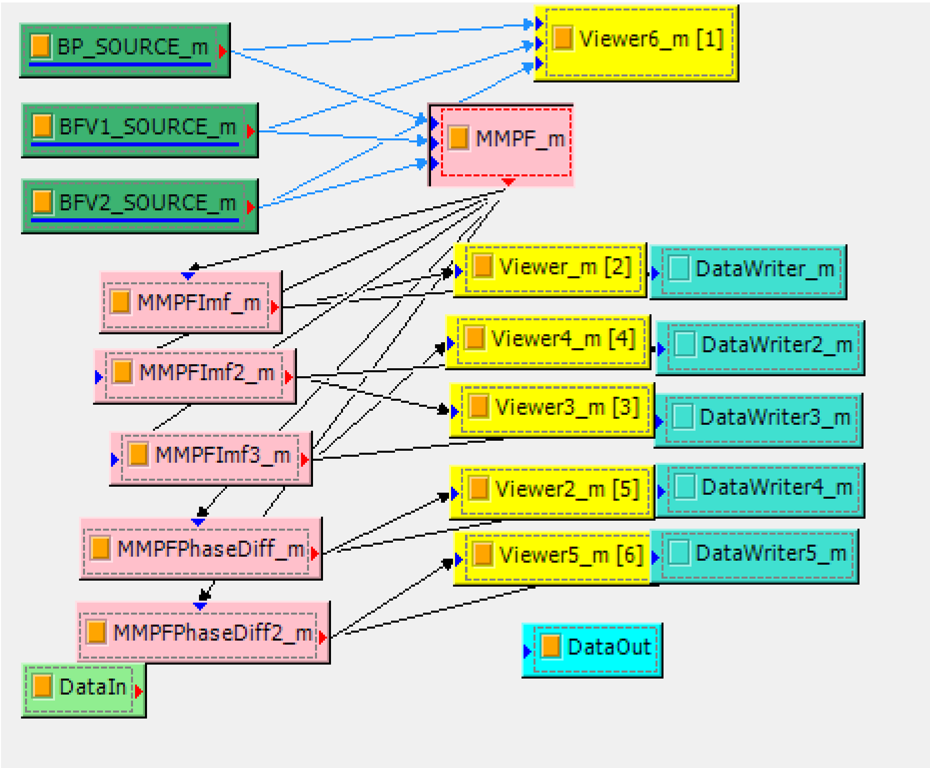
It's a user-defined macro.

Example

Use Compute-> MMPF-> MMPF Auto Macro, select input in following dialog.



BFV1 is the input of left BFV. If there is no right BFV, you can use BFV1 as the input of the right BFV. Set the parameters and the Output Directory, and then click "Continue". The Signal Flow Diagram between the SFOs is shown below.



By selecting Viewer_m SFO, the wave of signal is shown. If DataWriter_m SFO is selected, data is written in the Output Directory.

Related Functions

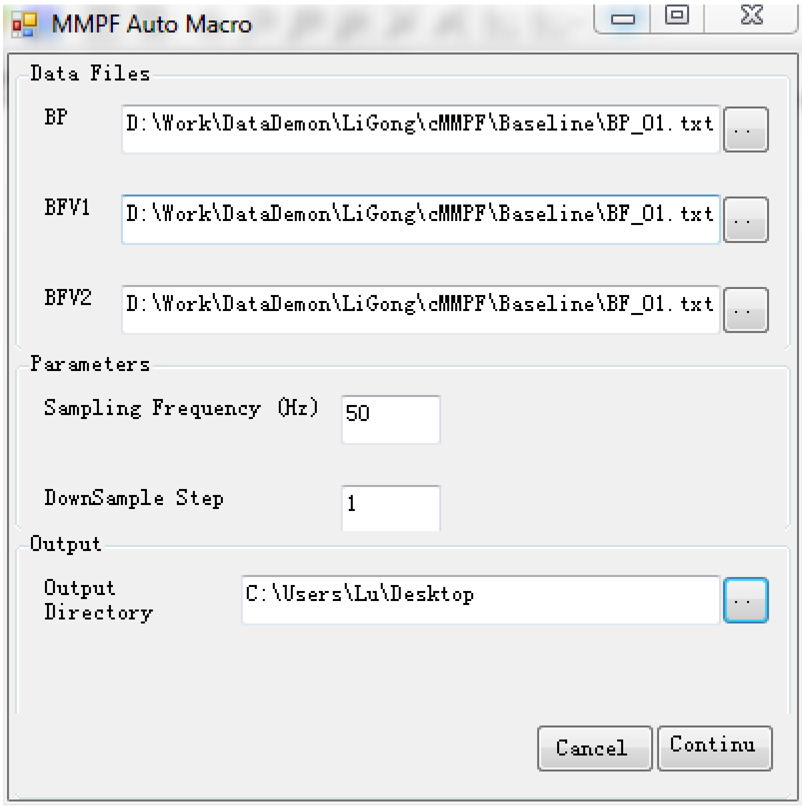
RCADA EEMD, Hilbert Transform

3.11.6 MMPF Expert Macro

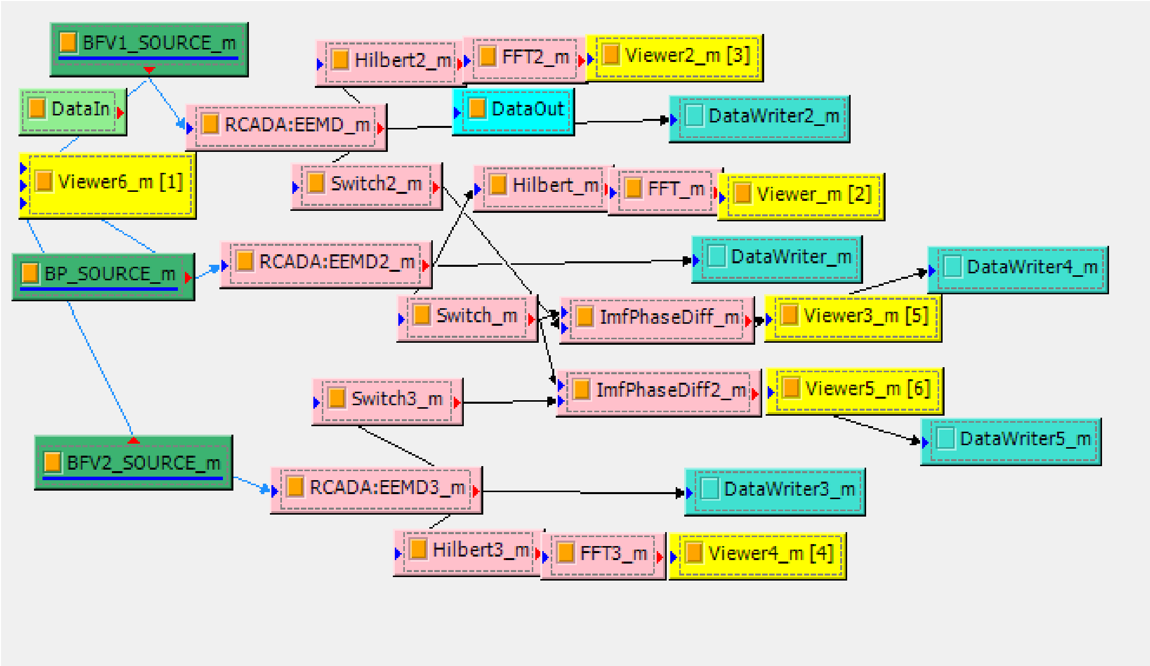
It's a user-defined macro.

Example

Use Compute-> MMPF-> MMPF Expert Macro, select input in following dialog.



BFV1 is the input of left BFV. If there is no right BFV, you can use BFV1 as the input of the right BFV. Set the parameters and the Output Directory, and then click "Continue". The Signal Flow Diagram between the SFOs is shown below.



By selecting Viewer_m SFO, the wave of signal is shown. If DataWriter_m SFO is selected, data is written in the Output directory. The parameters of RCADA: EEMD_m SFO can be modified and the corresponding IMFs can be selected.

Related Functions

RCADA EEMD, Hilbert Transform, FFT

Chapter 4: Format Conversion Signal Flow Object

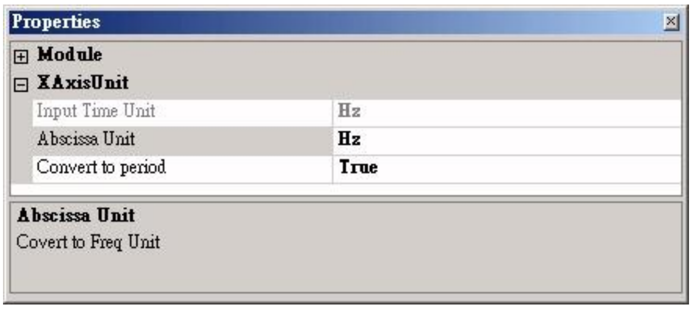
4.1 Change X Axis Unit

After reading in the signal data, the time unit of the data usually needs to be changed. In this case, *Change X-Axis Unit* can be used to convert time directly. In addition to time unit conversion, this module can also convert the spectrum-axis, i.e. the X-axis, from frequency to period.

Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular) and Audio (which could be real number or complex number, single channel or multi-channel, Regular). The output formats are real, complex, single channel/multiple channel Regular signal and audio signal. If the property of *Convert to period* is changed to True, the format of output signal would be changed from Regular to Indexed, because the data in x-axis are not separated with equal interval anymore.

Property of *Abscissa Unit* shows the unit of x-axis to be converted to. The defaults is in second. Changing the *Abscissa Unit* can trigger the unit conversion on x-axis for the input data. The explanation of the unit is given in the table below.



Property Name	Property Definition	Default Value
Convert to period	When the unit of x-axis is frequency, this option can convert the unit of frequency to the unit of period on x-axis.	**False**
ps	Picosecond	10^{-12} second
ns	Nanosecond	10^{-9} second
us	Microsecond	10^{-6} second
ms	Millisecond	10^{-3} second

sec	Second	1second
min	Minute	60 second
hour	Hour	60 minutes
day	Day	24 hours
week	Week	7 days
month	Month	30 days
year	year	365 days

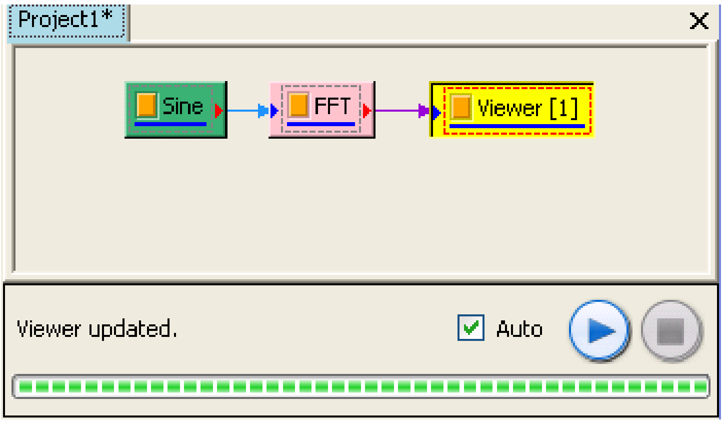
Property Name	**Property Definition**
THz	Terahertz Cycles per 10^{-12} second
GHz	Gigahertz Cycles per 10^{-9} second
MHz	Megahertz Cycles per 10^{-6} second
KHz	Kilohertz Cycles per 10^{-3} second
Hz	Hertz Cycles per second
Cycles_per_min	Cycles per minute
Cycles_per_hour	Cycles per hour
Cycles_per_day	Cycles per day
Cycles_per_week	Cycles per week
Cycles_per_month	Cycles per month
Cycles_per_year	Cycles per year

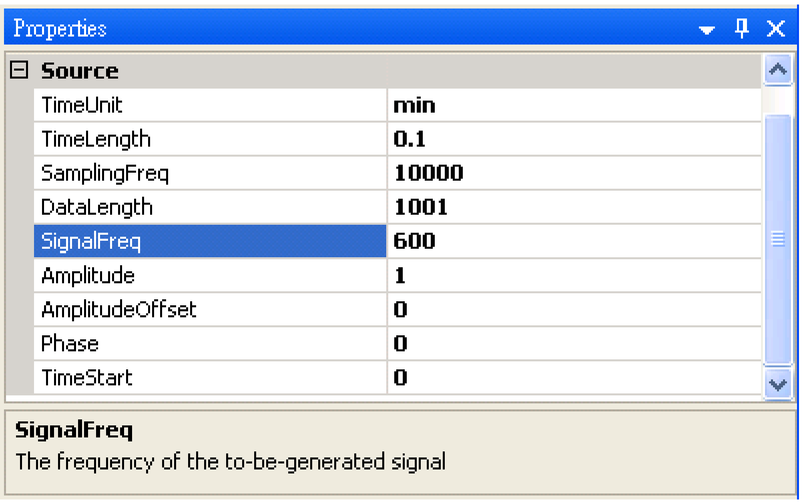
Example

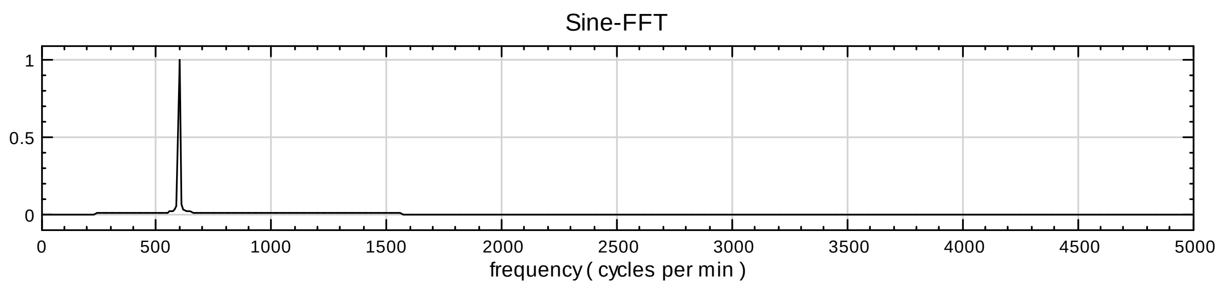
1. Select *Source*→*Sine Wave* to create a sine wave with default frequency of 10Hz,

sampling frequency of 1000Hz, and length is 1 second. Then change the *Properties/TimeUnit* to minute, *SamplingFreq* to 10000, *SignalFreq* to 600 for obtaining a signal whose x-axis unit is in minute and signal frequency is 10Hz.

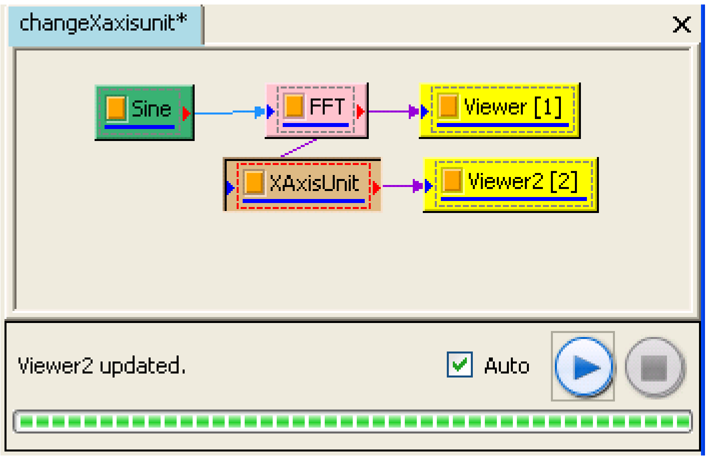
Connect the signal to *Compute* → *Transform* → *Fourier Transform* for FFT calculation, and then connect the Viewer to show the curve, where the x-axis is in frequency and the unit is in cycles per minute.

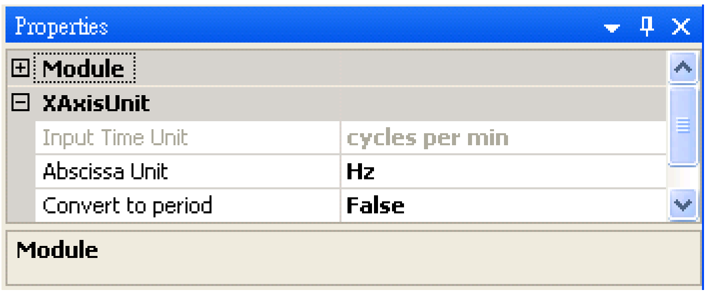


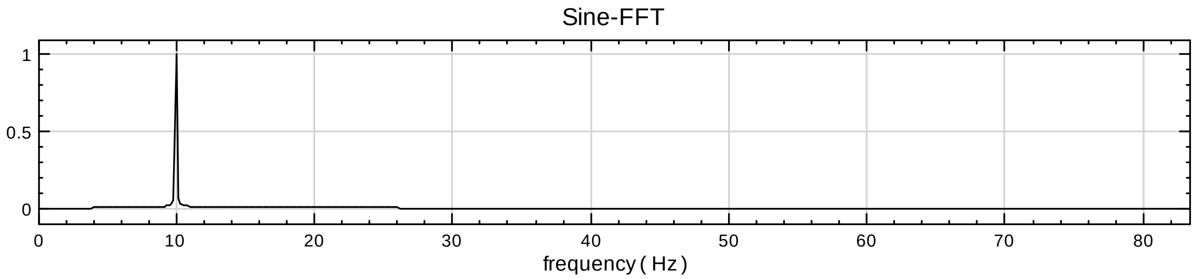




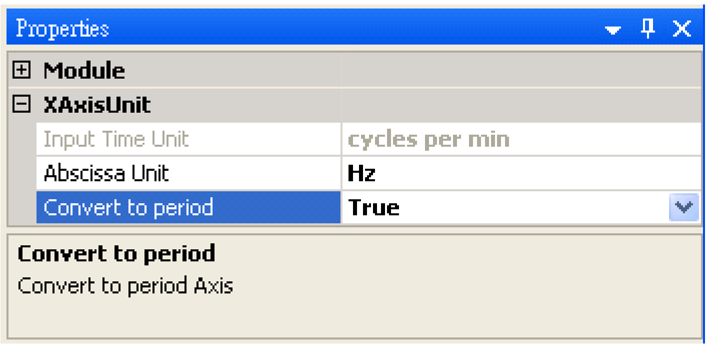
From the result, the frequency is 600 cycles per minute. It is not in Hz. Connect Change X Axis Unit to the output of FFT, change the *Properties/*Abscissa unit and use Channel Viewer to show the result. Now the x-axis has changed to Hz and the values on x-axis has also changed to second automatically.

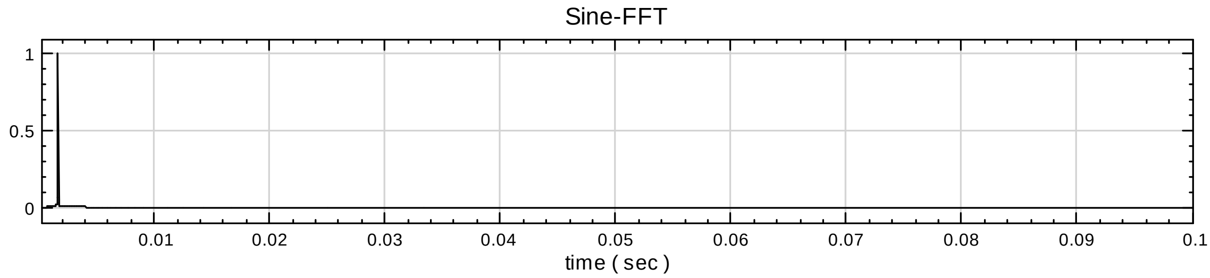






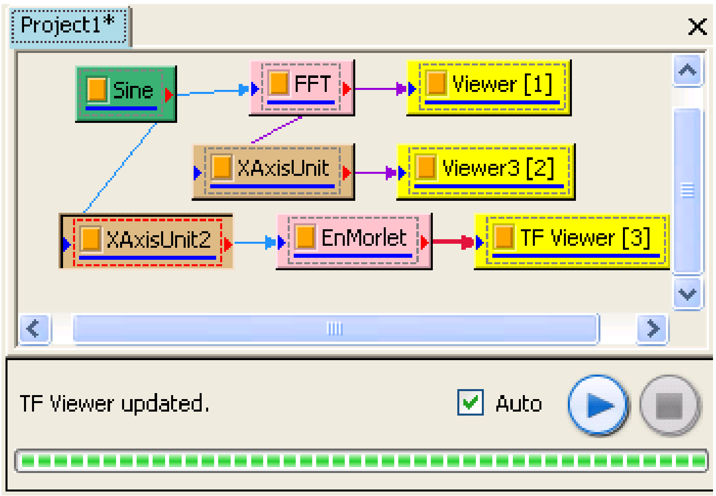
2. In addition, the *Properties/Convert to period* can be set to True. This converts the x-axis from frequency to period, as shown in the figure below. The x-axis is converted to Time and the unit is in second, i.e. the period corresponding to Hz.

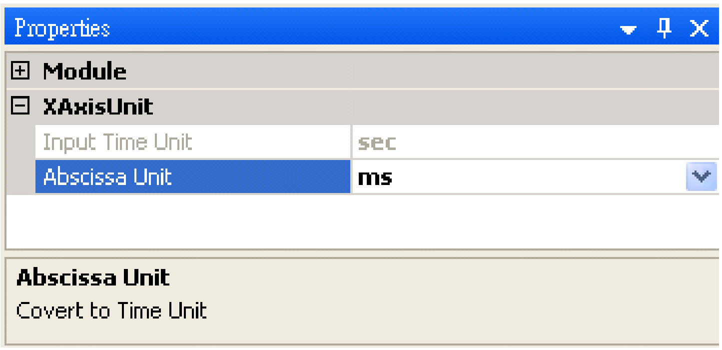


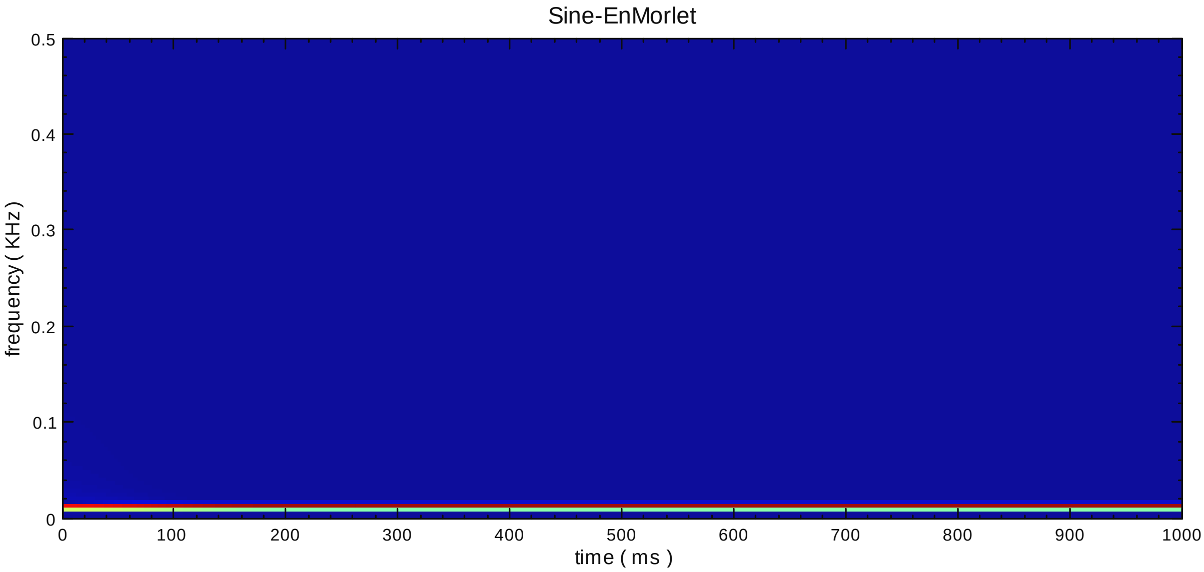


3. The time-frequency analysis module (TFA) is not able to pass the result to *Change X Axis Unit* for frequency unit modification directly, since the frequency is located on the y-axis fo the time-frequency diagram. However, the conversion can be achieved by changing the unit of x-axis first and then performing time-frequency analysis.

   Connect *Change X axis Unit* to the source signal of sine wave, change the *Properties/Abscissa unit* to msec, use *Compute* → *TFA* → *Enhanced Morlet Transform* to perform time-frequency analysis, and then use *Viewer* → *Time-Frequency Viewer* to plot the result. It shows that the y-axis, i.e. the frequency axis, has been changed to KHz, i.e. 1/msec.







Related Functions

Import Data from File, Viewer, Fourier Transform, Enhanced Morlet Transform.

4.2 Convert to Audio

This component changes the type of signal data from Signal to Audio.

Introduction

The output data format of *Convert to Audio* follows the Microsoft Wave Format. The Microsoft Wave Format contains 3 data blocks: RIFF, FMT, and DATA. The details are given as follows.

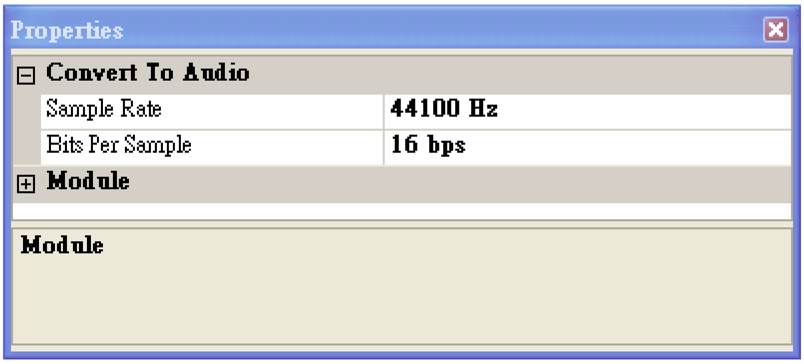
RIFF: defines the file format, file size and other information. The format is WAVE.

FMT: contains the related properties of audio signal such as code type, sampling frequency, number of audio channels, byte rate etc.

DATA: The original data which contains audio information.

Properties

Acceptable input data sources are: real, single channel or multi-channel, Regular signal or audio signal. Note that this component only accepts double channels for multi-channel. The output format is real, single channel or double channel, Regular audio signal. Available properties are *Sample rate* and *Bits per sample*.



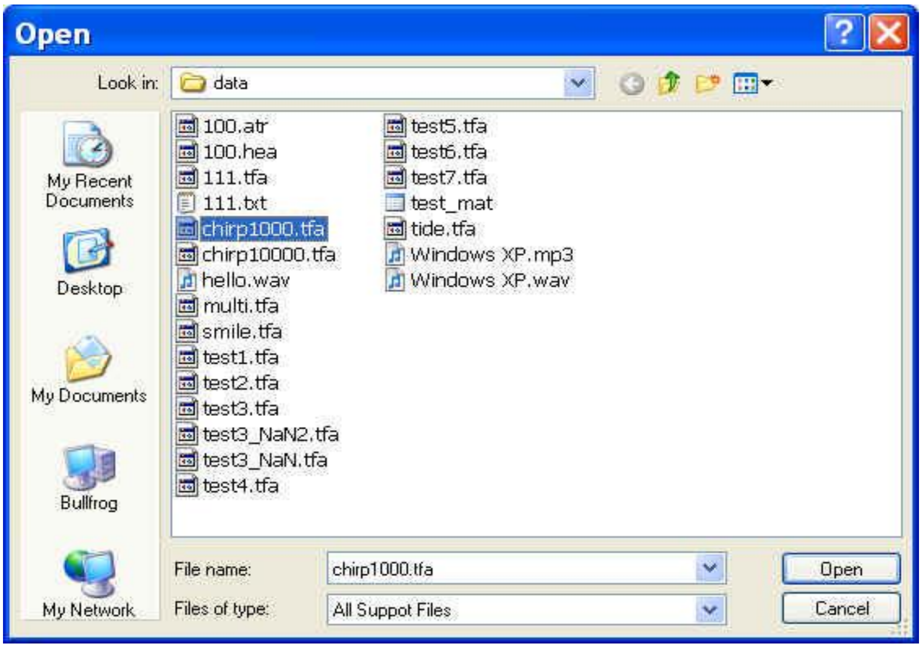
Property Name	Property Definition	Default Value
Sample Rate	The number of sampling points in every second. It affects the resolution of voice frequency. The available options are 1000, 2000, 4000, 8000, 11025, 22050, 44100, 48000, and 96000Hz.	44100

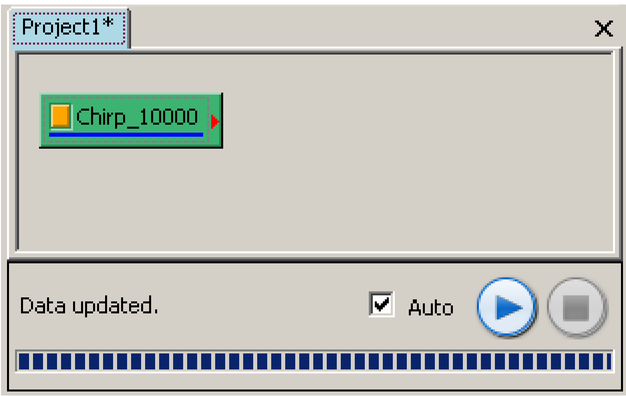
Bits Per Sample	Define the value of every saved data which could affect the resolution of sound intensity. The available options are 8, 16, 24, and 32 bps.	16

Example

Convert a signal data file “Chirp1000.tfa” to audio signal using *Convert To Audio.*

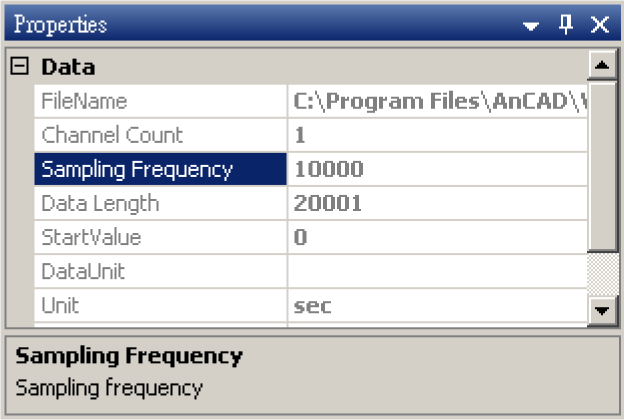
1. Press in the Network tools, or use *Source→Import data from File* to read the signal file, chirp1000.tfa, in the data folder of installation directory which has a default location *C:\Program Files\DynaDx\DataDemon\data.*





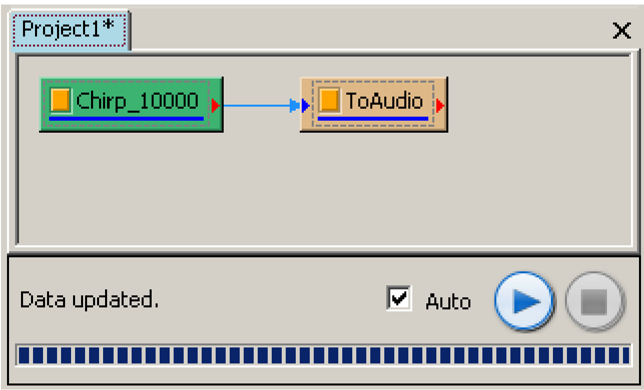
In Properties, it shows that this signal has the SamplingFrequency of 10000, the

DataLength of 20001 and the Unit in second.



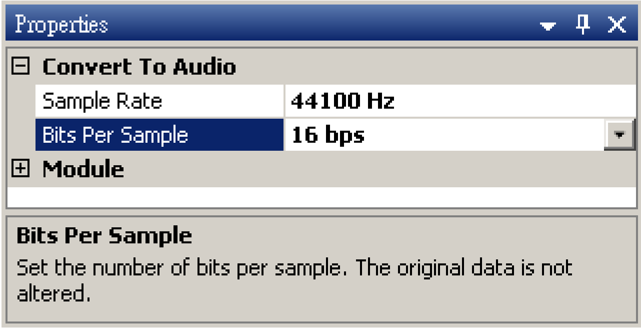
In addition, open the Module type in the Properties, where the OutputDataType shows the signal format and type of this module output. Since the OutputDataType is *Real Single-Channel Signal of Rank-1(Regular) Data*, the data type of Chirp1000 is Signal. Refer to the introduction of Properties in Chapter one for more details.

2. From Chirp10000 SFO, select *Conversion→Convert To Audio* directly.



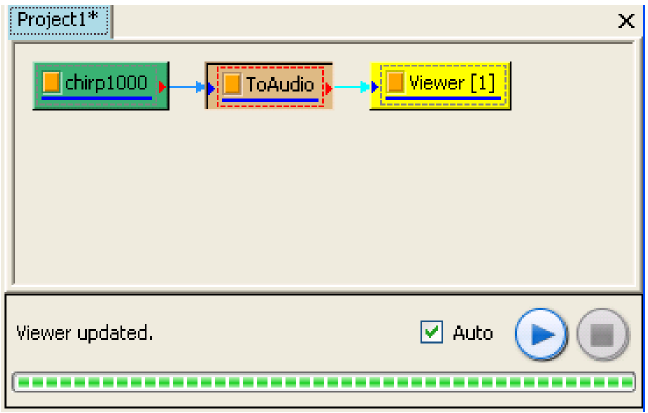
In Properties, it can be seen that the Sample rate = 44100Hz and Bits Per

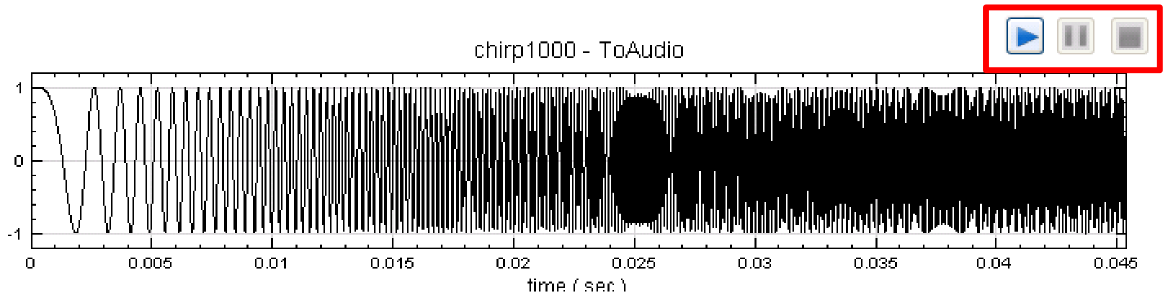
Sample = 16bps. These two properties can be changed in the drop-down menu.



Then check Module in Properties and it shows that the OutputType has been changed to Audio.

3. Connect *Viewer→Channel Viewer* to *ToAudio*, the tool at the top right of the Viewer could be used to play this audio signal.





Related Functions

Channel viewer, Wave Writer.

References

Microsoft Wave Format：

http://ccrma.stanford.edu/CCRMA/Courses/422/projects/WaveFormat

4.3 Convert to Regular

This component changes the time-axis setting of a signal from Indexed to Regular.

Introduction

When reading text files such as .txt and .csv etc with *Import data from file*, if a row or a column in the data contains time-axis coordinates, it is recommended to use the *Specify Time column/row* in the *Text Importer*. As such, the data format is marked as Indexed and the sampling periods are assumed to be different. However, most modules require the signal format to be Regular. This module can be used to convert Indexed signals to Regular signals.

In Indexed format, it is assumed that the data points on time-axis are discrete and the intervals are uneven. Therefore, there exists a corresponding time coordinate for every data point. Let the input signal be $X^{(I)} = \left\{x_0^{(I)}, x_1^{(I)}, \cdots x_{N-1}^{(I)}\right\}$, and the signal time-axis is defined as,

$$T^{(I)} = \left\{t_0^{(I)}, t_1^{(I)}, \cdots t_{N-1}^{(I)}\right\}$$

Convert to Regular performs re-sampling on the signal above to convert the time-axis of Indexed signal into Regular which is discrete and equidistant.

$$t_j^{(O)} = t_0^{(O)} + j \times \Delta t^{(O)} \text{，} 1 \leq j \leq M - 1$$

where $t_j^{(O)}$ is the time-axis after *Convert to Regular* processing, $\Delta t^{(O)}$ is the output sampling period, *M* is the number of the output data points. The output signal $X^{(O)}$ is obtained by the formula below.

$$X^{(O)} = \left\{x_0^{(O)}, x_1^{(O)}, \cdots x_{M-1}^{(O)}\right\}$$

Two types of calculations, FillGap and RemoveGap, can be used to convert Indexed to Regular. The details are given below.

FillGap:

FillGap can preserve the signal time characteristics and add values at the locations where the time intervals are too large. In calculation, it can detect the minimum sample period, $\Delta t_{\min}^{(I)}$, in the input signals, and then use it to perform re-

sampling on the signals, $\Delta t^{(O)} = \Delta t_{\min}^{(I)}$. The re-sampling methods are the same as the ones in the Resample component with 7 methods. Users can also set sampling period manually. However some constraints may apply. For consistency of input and output signals, the sampling period $\Delta t^{(o)}$ must be less than or equal to 1.5 times of $\Delta t_{\min}^{(I)}$. The computation logics of re-sampling are briefly introduced as follows.

If $t_{j+1}^{(I)} - t_j^{(I)} > 1.5 \times Min(\Delta t^{(o)})$, the following methods are used to calculate the filling data, $x_j^{(o)}$, between $x_j^{(I)}$ and $x_{j+1}^{(I)}$.

Fix: Use fixed value as filling-value.

Prev: Use value of the preceding point as filling-value.

Next: Use value of the subsequent point as filling-value.

Linear Interpolation: Use the preceding and subsequent points to perform l inear Interpolation

Spline Interpolation: Use Spline to fill values.

Monotonic Cubic Spline: This is a 3-degree interpolation with damping. It has better performance than Spline in the case of processing signal with large slope like square wave because it can avoid large vibration.

No Fill: No additional value is added, NaN.

If $t_{j+1}^{(I)} - t_j^{(I)} < 1.5 \times Min(\Delta t^{(o)})$, the value which is corresponding to the input signal of $x_{j+1}^{(I)}$ would be output directly to the corresponding position, $X^{(o)}$, in the output signal.

RemoveFillGap:

RemoveGap discards the time-axis $T^{(I)}$ of the input signal, uses the starting time $t_0^{(I)}$ and the minimum sampling period $\Delta t_{\min}^{(I)}$ to re-calculate the time-axis $T^{(O)}$ of the output signal, and replaces the time-axis $T^{(I)}$ with $T^{(O)}$.

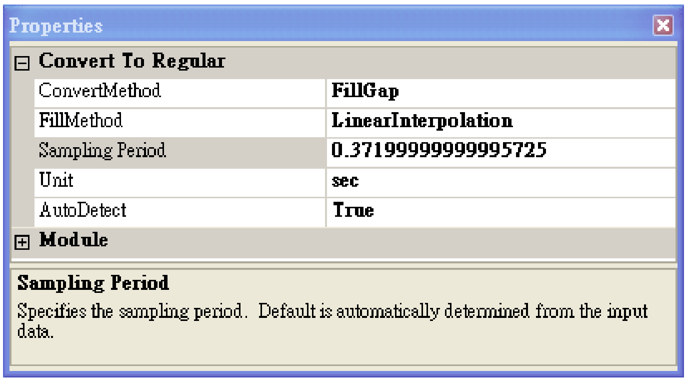
The formula for $T^{(O)}$ is

$$t_j^{(O)} = t_0^{(I)} + j \times \Delta t^{(O)},\ 1 \le j \le N-1$$

Therefore, the output signal from RemoveGap has the same number of data points as the input signal. However, uneven intervals in the input signal are changed to even intervals.

Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Indexed). The outputs are Regular signals which are real or complex, single channel or multiple channels.



Property of AutoDetect provides the option to set *Sampling Period* of the output signal manually. If it is set as True, this module would detect the minimum sampling period of the input signals automatically and use it as the Sampling Period. If AutoDetect is set as False, user can set the Sampling Period manually. Because a large sampling period would cause discrepancies between the output signal and the original one, the manual setting of the Sampling period must be less that 1.5 times of the sampling period obtained by AutoDetect.

Property of IConvertMethod allows users to select FillGap or RemoveGap to calculate the time-axis of the input signal. If FillGap is selected, a new property of FillMethod is provided for value-filling methods, which are explained below.

Property Name	Property Definition	Default Value
Convert Method	FillGap: Use to conduct signal re-sampling by value filling RemoveGap: Directly change the time-axis of the original signal. Use the time starting point and the Sampling period to re-arrange data time	FillGap

FillMethod	When ConvertMethod = FillGap, different data-filling methods can be selected. FixedValue: Use NullValue as the fixed filling data PrevValue: Previous value NextValue: Next Value LinearInterpolation: Linear interpolation SplineInterpolation: Use Spline Curve to calculate the difference Monotonic Cubic: This is a 3-degree interpolation with damping. It has better performance than Spline in case of processing signal with large slope like square wave. NoFill: The value in this location is Null. No value is added.	LinearInterpolation
Sampling Period	To show or set the sampling period, $\Delta t^{(o)}$, of the output signal. When AutoDetect is set to True, it shows the minimum sampling period detected of the input signals, i.e. $\Delta t^{(o)} = \Delta t_{\min}^{(I)}$ When AutoDetect is set to False, besides showing $\Delta t_{\min}^{(I)}$, it can also be used to set the sampling period $\Delta t^{(o)}$.	$\Delta t^{(o)} = Min(\Delta t^{(I)})$
Unit	To show or set the sampling time unit of output signal. When AutoDetect is set to True, show the signal time unit detected. When AutoDetect is set to False, besides showing the signal time unit, set the signal time unit by using Sampling Period together.	Based on input signals

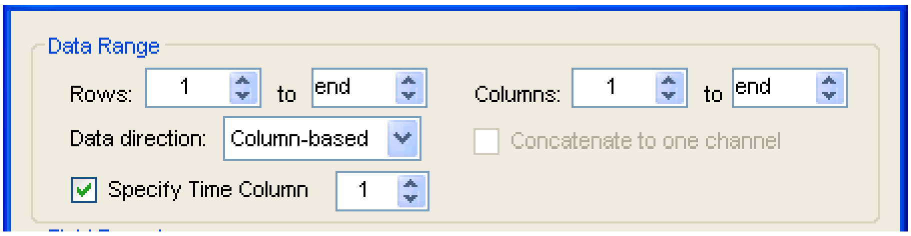
Property Name	Property Definition	Default Value
AutoDetect	Determine whether to detect Sampling Period and Unit automatically	True
NullValue	If ConvertMethod is set as FillGap and FillMethod set as FixedValue, this property would be provided to set the fixed value for data-filling	0

Example

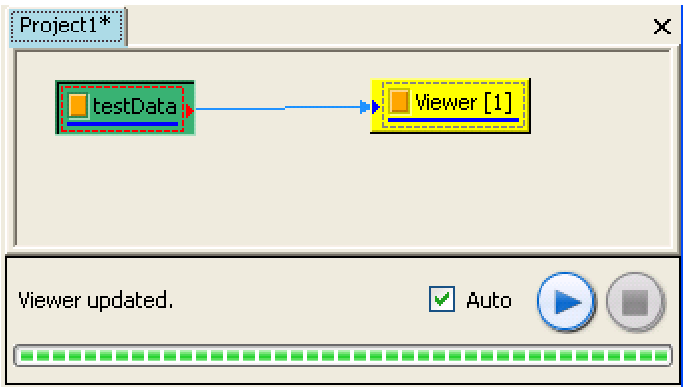
1. Read one set of signal data which are in Indexed format. First, generate one set of simple data as shown below, where the first column is time while the second column is data.

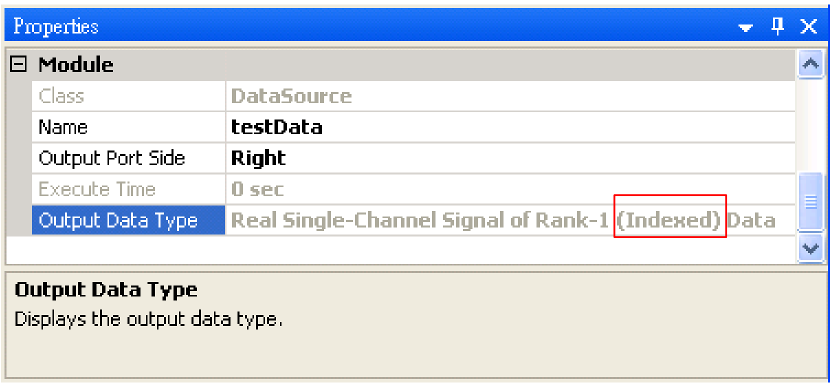
```
0      1
0.1    2
0.2    3
0.3    4
0.5    5
0.8    6
1.3    7
2.1    8
3.4    9
5.5    10
```

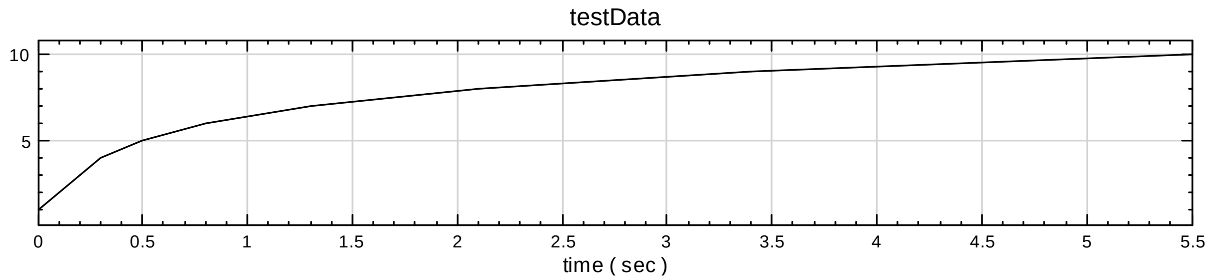
Then, press the in the Network tools or use Source→Open Data to read a signal file, TestData.txt. Check the *Specify Time Column* in *Text Importer* and then press OK.



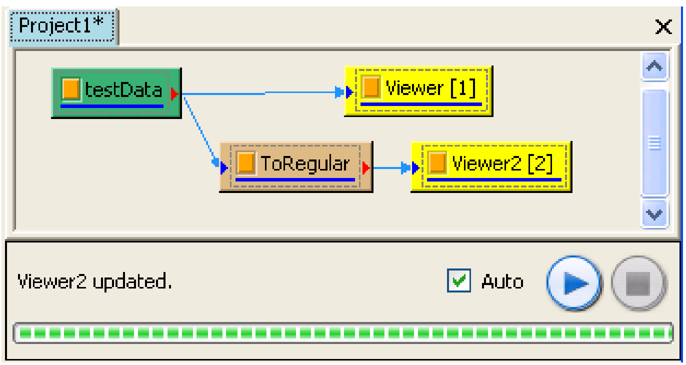
After reading the signal, use *Viewer→Channel Viewer* to plot figures. And select TestData to verify the OutputDataType in *Properties/*Module. It can be seen that the time-axis format is Indexed.

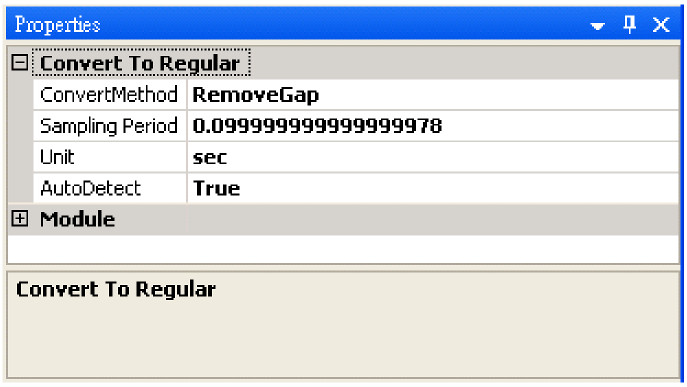


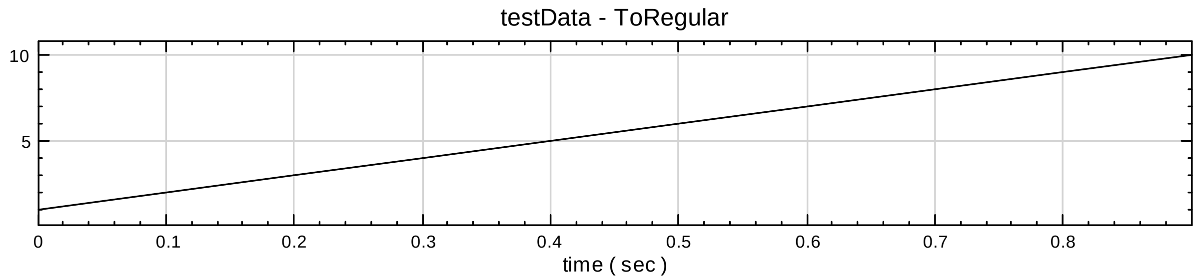




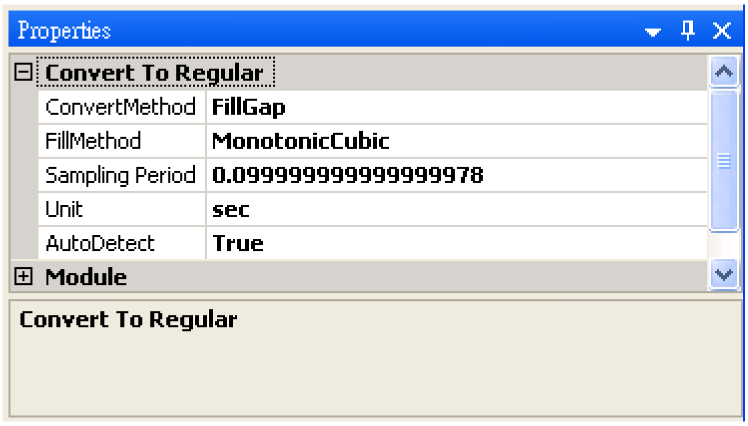
2. Connect *ToRegular* to *TestData* for converting the signal into an evenly sampled data and then use *Channel Viewer* to plot the result. In the Properties of ToRegular, it shows that the default method is RemoveGap. The Sampling Period detects the minimum sample period and this value is used for re-sampling. Therefore, the sampling period is 0.1 second and the total time length is 0.1 x 9 = 0.9 second.

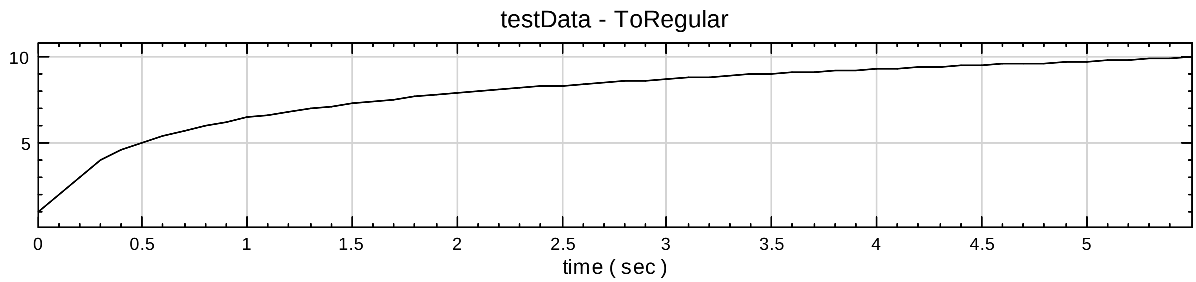






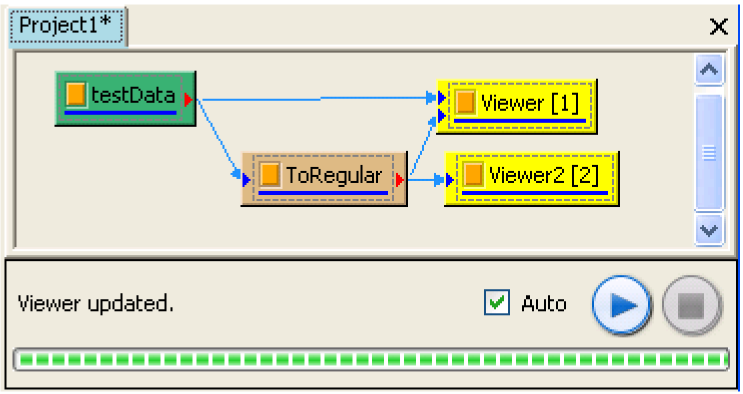
3. Change the *Properties/ConvertMethod* in ToRegular to FillGap, the FillMethod to Monotonic Cubic. The output result is shown as below. It shows that the FillGap preserves the time-axis definition of the original signal and the signal time-axis is changed to even interval of an approximate 0.1 second sampling frequency.

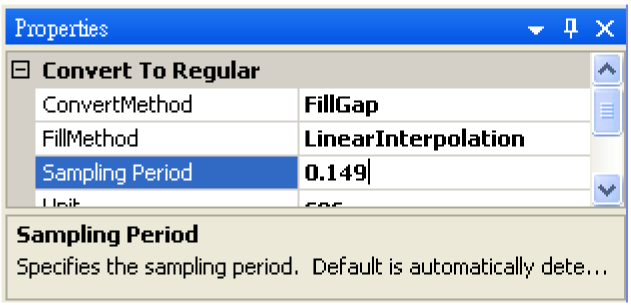


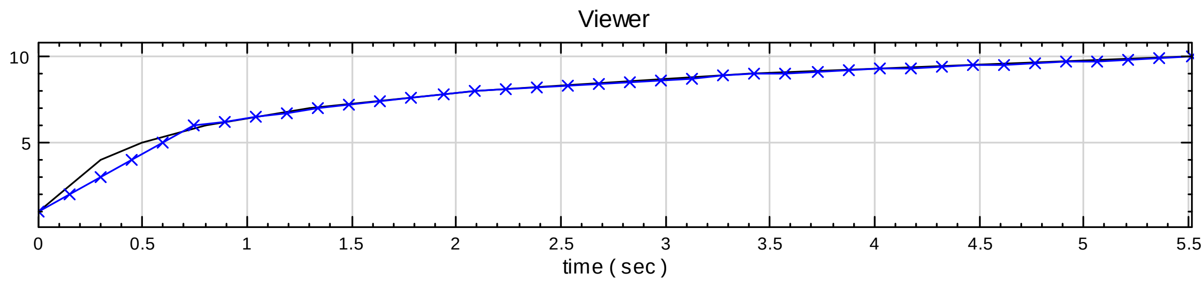


4. ToReguar allows the users to fine tune the signal sampling frequency. First, set AutoDetect as False, and then change the SamplingPeriod to 0.149. Dragging the

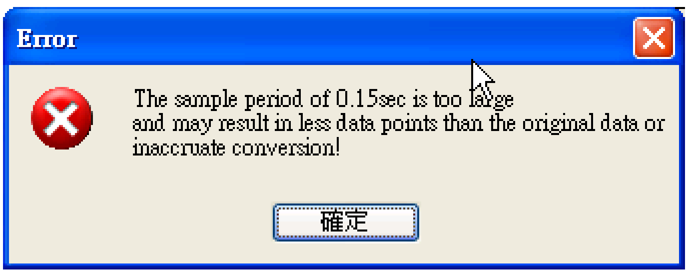
output result to Viewer[1] to compare with the original signal, where the black curve is the original signal and the blue curve with「-x」is the ToRegular signal. It shows that the signal with bigger sampling frequency is distorted.

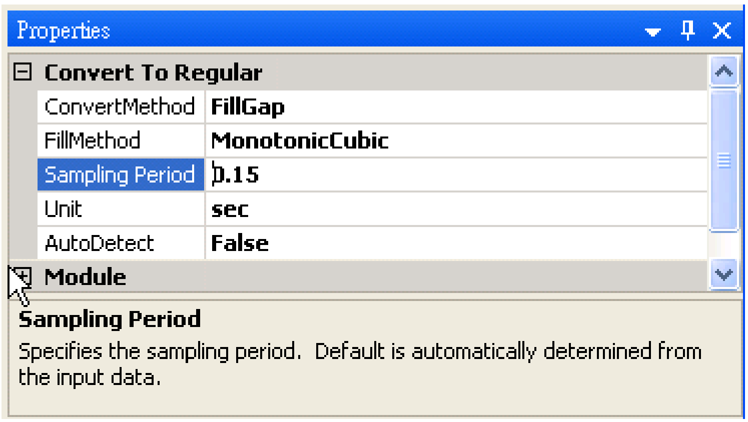






5. Next, try to change the Sampling Period to 0.15. An error message is popped up for not allowing to enter a value which is bigger than 1.5 times of the minimum sampling frequency of the input signal.





Related Functions

Convert to Audio, Filling Null Value, Resample.

4.4 Map to Real

This component converts a Complex signal to a specific Real Signal.

Introduction

Let $X^{(c)} = \left\{x_0^{(c)}, x_1^{(c)}, \cdots, x_{N-1}^{(c)}\right\}$ and $Y^{(c)} = \left\{y_0^{(c)}, y_1^{(c)}, \cdots, y_{N-1}^{(c)}\right\}$ to be the real part and the imaginary part of a complex signal, respectively; where c represents the cth channel. The output signal $Z^{(c)}$ can be used to calculate real signals. There are 6 types as shown below.

$$z_j^{(c)} = \sqrt{\left(x_j^{(c)}\right)^2 + \left(y_j^{(c)}\right)^2} e^{i\theta} = Ae^{i\theta}$$

Magnitude: A

Phase: θ

Real Part: $x_j^{(c)}$

Imagine Part: $y_j^{(c)}$

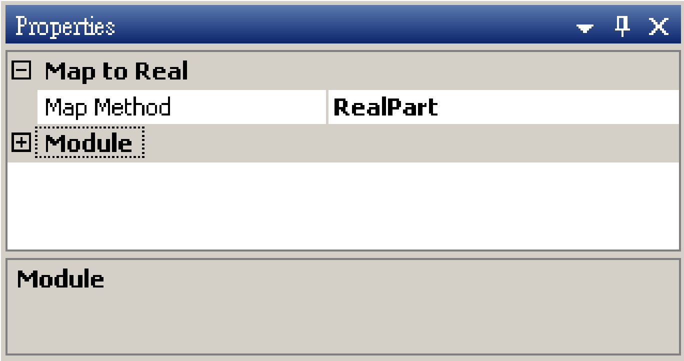
Gain: $20 \times \log(\frac{A}{Gainref})$, Gainref is the Gain reference

Power Spectrum: A^2

Properties

This module accepts input of Signal (which could be complex number, single channel or multi-channel, Regular or Indexed), Audio (which could be complex number, single channel or multi-channel, Regular), Numeric (which could be complex number, single channel or multi-channel, Regular or Indexed) and Spectra (which could be complex number, single channel or multi-channel, Regular). The output format is identical to the input signal except that it is a real number.

Property is *Map Method* with a default value of Real Part, i.e., the real part of the input signal. *Imag Part* represents the imaginary part, Magnitude is the absolute value of the complex signal, Phase denotes the phase, Gain is used to set *Gain Reference* for Power Gain calculation, and Power Spectrum is the power of Magnitude.

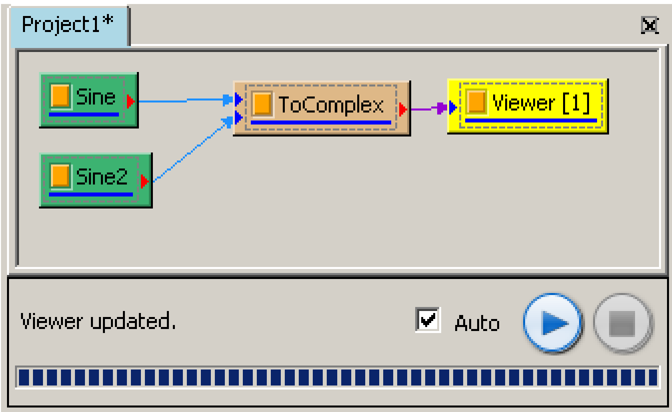


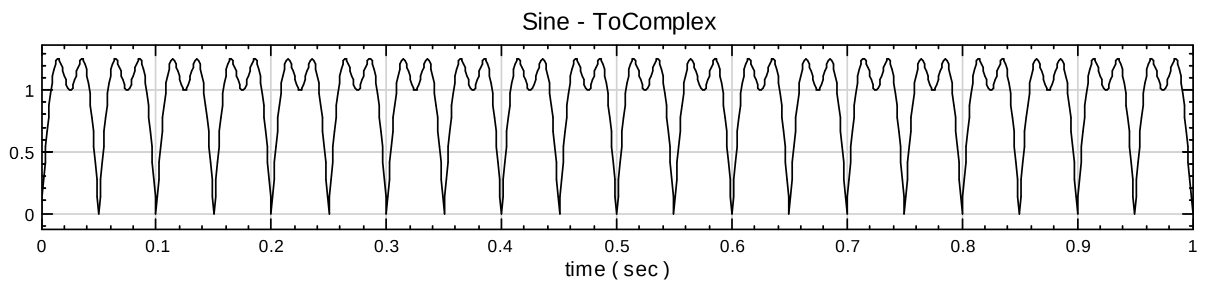
Property Name	Property Definition	Default Value
MapMethod	Select the signal type which the complex signal should be converted to. The method options are Magnitude, Phase, RealPart, ImagPart, Gain, and Powerspectrum.	RealPart

Example

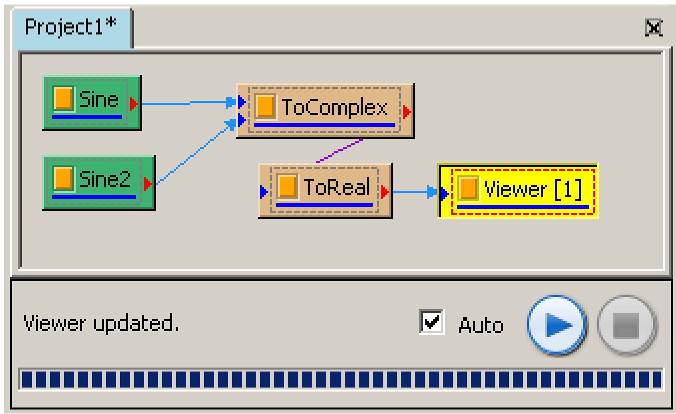
Two sine waves with the sampling frequency of 1000Hz, length of 1 second and amplitude of 1, are used as input signals.

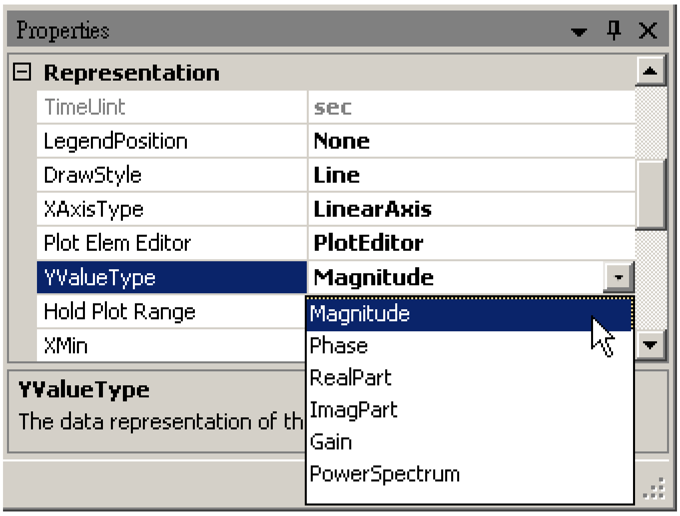
1. In the Network Window, use the Source→Sine to create two sine waves and then change their Property→SignalFreq to 10Hz and 20Hz respectively. Then, use *Compute→Conversion→Merge to Complex* to merge these two signals to a complex signal and use *Viewer→Channel Viewer* to plot the result.

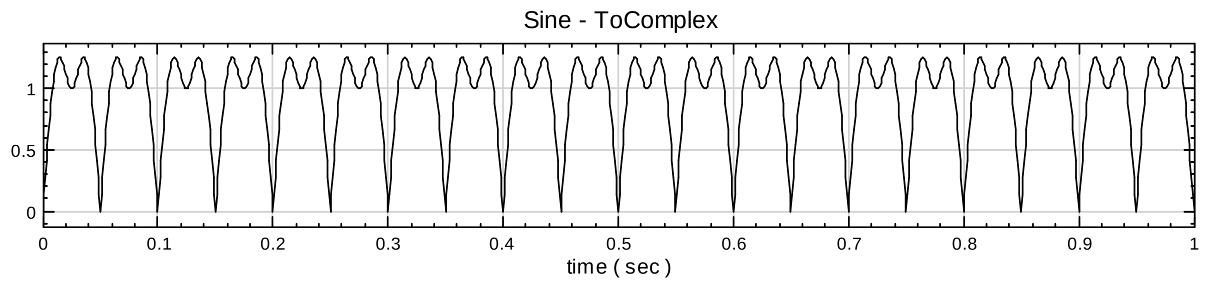




2. Use *Map to Real* to convert this complex signal to a real signal. Change the Map Method in the Properties and then use *Viewer→Channel Viewer* to show the result in the window. The default output of *Map to Real* is the signal Magnitude.







Note that although the computation procedure is different, the results shown in

Channel Viewer are identical. For the 1st one, it is obtained by calculating the Magnitude of the complex signal by *Channel Viewer*; for the 2nd one, Magnitude is obtained by *Map To Real* then plotted by *Channel Viewer.*

Related Functions

Sine Wave, Merge to Complex, Channel Viewer.

4.5 Merge to Complex

This component merges two real signals to form a complex signal. The real part of the complex signal is the 1st input and the imaginary part is the 2nd input signal.

Instruction

Let $X^{(cx)} = \{x_j^{(cx)}\}$ to be a multi-channel signal, where *cx* represents the channel, *j* is the signal time axis. Also let $Y^{(cy)} = \{y_k^{(cy)}\}$ to be another signal, where *k* is the signal time axis. The number of channels and the data lengths of these two signals may be different. The output signal of *Merge to Complex* for these two signals is

$$z_m^{(c\_ref)} = x_m^{(c\_ref)} + i\, y_m^{(c\_ref)}$$

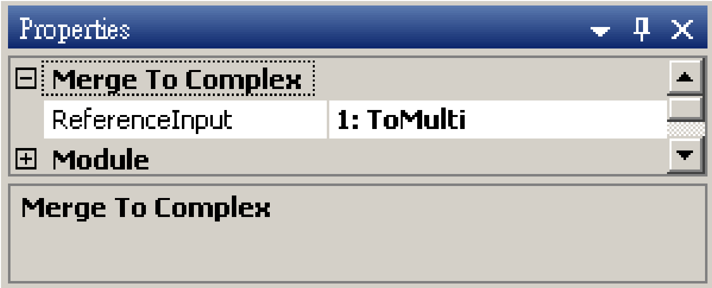
where c_ref and m represent the number of channels and data length respectively. If the reference signal is $X^{(cx)}$, we have $c\_ref = cx$, $m = j$. The sampling frequencies of the resulting complex signal is identical to $X^{(cx)}$.

Properties

This module accepts input of Signal (which could be real number, single channel or multi-channel, Regular or Indexed), Audio (which could be real number, single channel or multi-channel, Regular), and Numeric (which could be real number, single channel or multi-channel, Regular or Indexed).

The property of *Reference Input* is used in case when the number of channels of input signals is different or the time-axis is different. The users need to select one input signal as time axis reference for the output signal. The default value is 0 which means that the output reference is the 1st input signal. The time setting, such as Sample Frequency and Time Shift of the 2nd input signal, is set to those in the 1st input channel. The principle of copying time-axis setting is that the time points of missing data is filled with 0 and time points of exceeding the time reference is discarded.

In order to avoid the operation confusion, it is recommended to use two signals that have the same number of channels and the same time-axis.



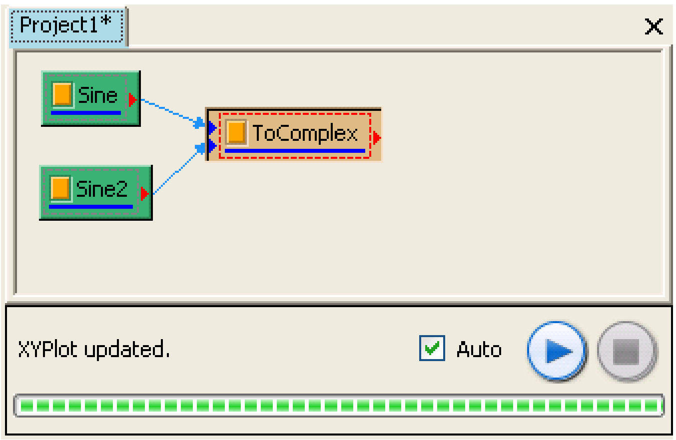
Property Name	Property Definition	Default value
ReferenceInput	Reference signal. Use the real or imaginary part as the reference for time-axis and number of channels	The 1st set of input signals

Example

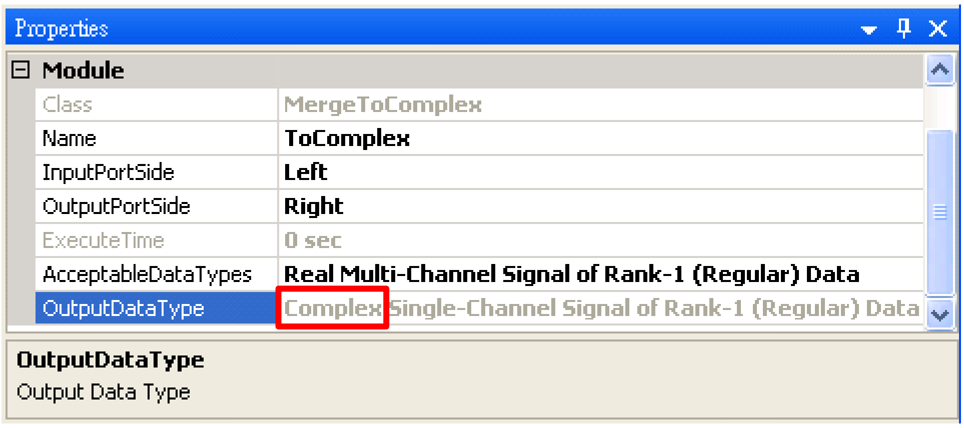
This example demonstrates two operation modes. First, merge two single-channel real signals to form a single-channel complex signal and show how to plot it in a complex plane. Second, use *Merge to Complex* to combine two signals which have the same sampling frequency but different time length to form a complex, multi-channel signal.

Single channel signal

1. Use *Source→Sine Wave* to generate two sine signals. Click the Sine2 icon and then change its *Properties/Phase* to 90 to make it a cosine signal. Next, use *Conversion→Merge to Complex* to merge these two signals into a complex signal.



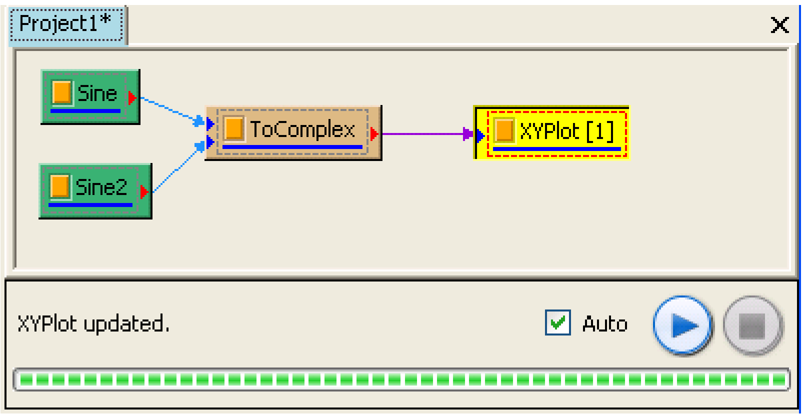
The output signal format of ToComplex is available in *Properties/*Module→OutputDataType. It has been changed to Complex.

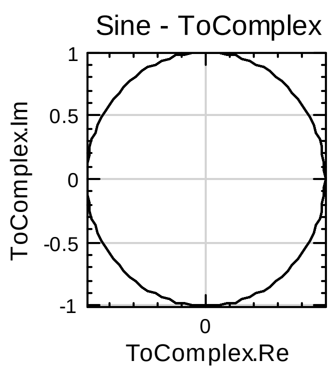


2. Connect *Viewer→XY Plot Viewer* to ToComplex. The viewer uses the real part of input signal as X-axis value and the imaginary part as Y-axis value to plot the signal following the time order.

$$Z_t = \sin(\frac{2\pi}{180}\cdot 10\cdot t) + i\cos(\frac{2\pi}{180}\cdot 10\cdot t)$$

Therefore, the signal can be plotted in the complex plane.

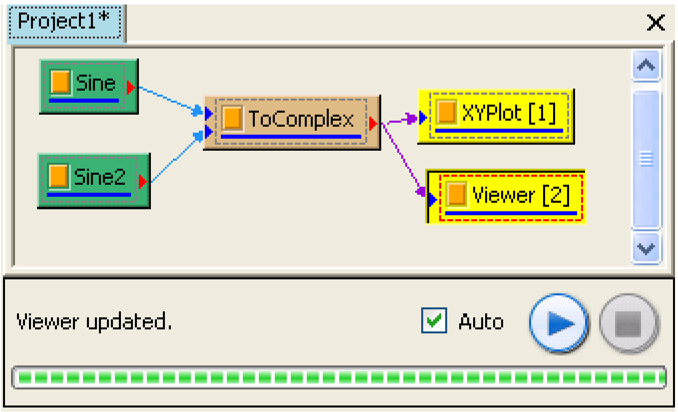


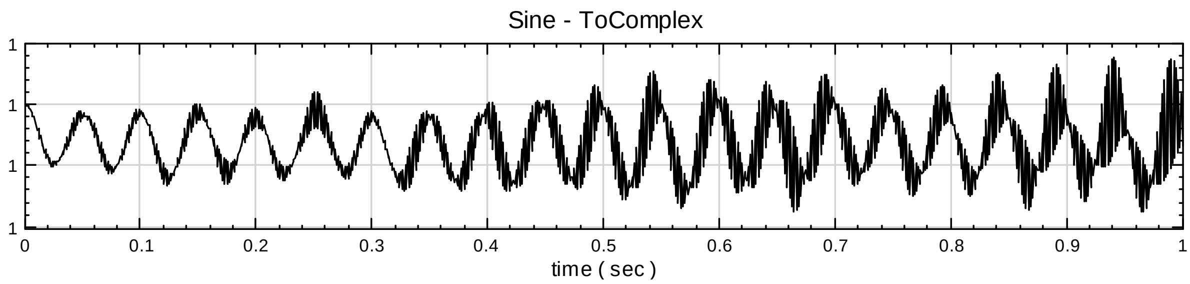


The figure above is obtained by making the *Properties/ViewerHeight* to be the

same as the *ViewerWidth* in the *XY Plot Viewer*. So the X-axis and Y-axis have identical scale. For detailed introduction about *XY Plot Viewer* operation, please reference to chapter of XY Plot Viewer.

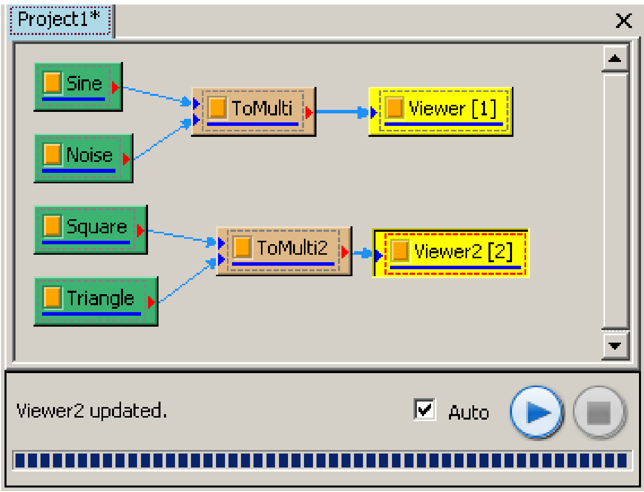
3. With *Channel Viewer*, timing diagrams with different computation cost can be obtained by modifying the *Properties/*YValueType. The figure below shows the Magnitude of this complex signal.

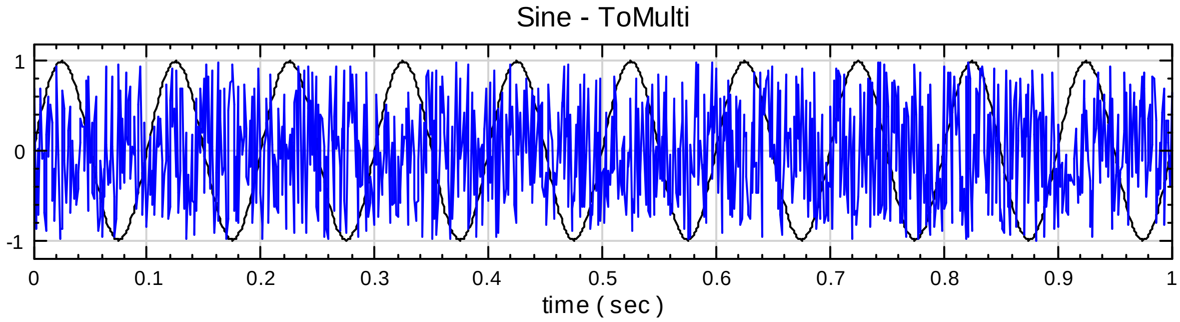




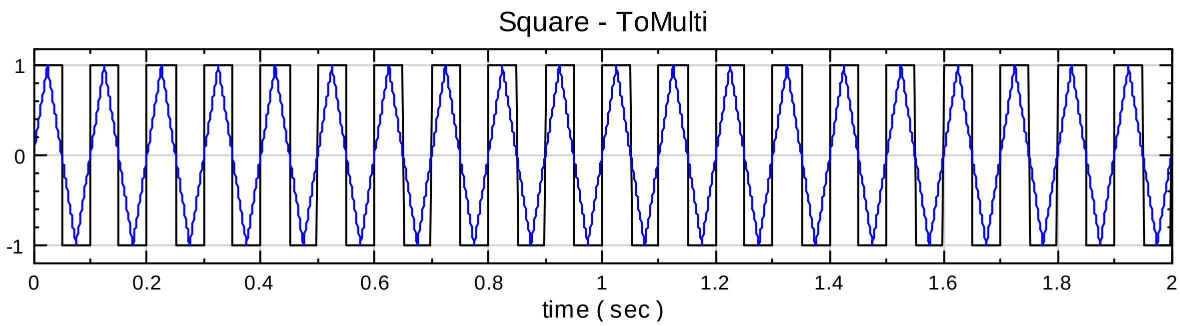
Multi-Channel signal

1. In Network Window, select Source→Noise to generate a white-noise signal and Source→Sine to generate a since Wave with default settings. Then, use *Computer→Conversion→Merge to Multi-channel* to merge these two signals into a 2-channel signal. Follow the same procedure to merge a square wave and a triangular wave into a 2-channel signal. And set the *Properties/*TimeLength to 2 seconds. And use *Viewer→Channel Viewer* to show the result.



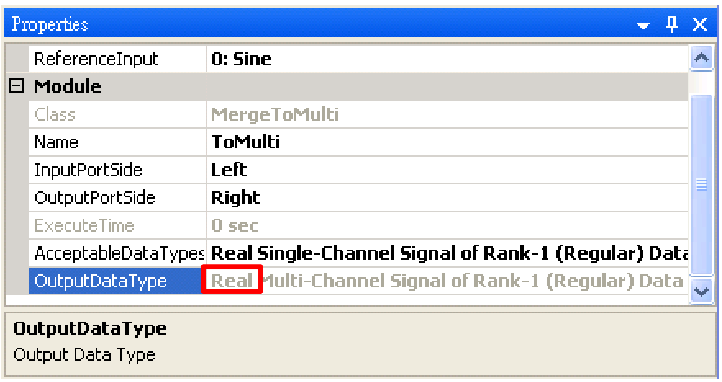


Signal diagram in ToMulti (sine wave and white noise)

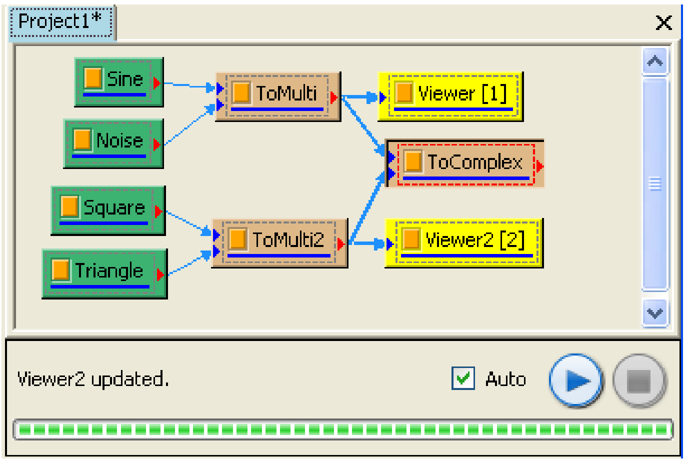


Signal diagram in ToMulti2 (square wave and triangle wave)

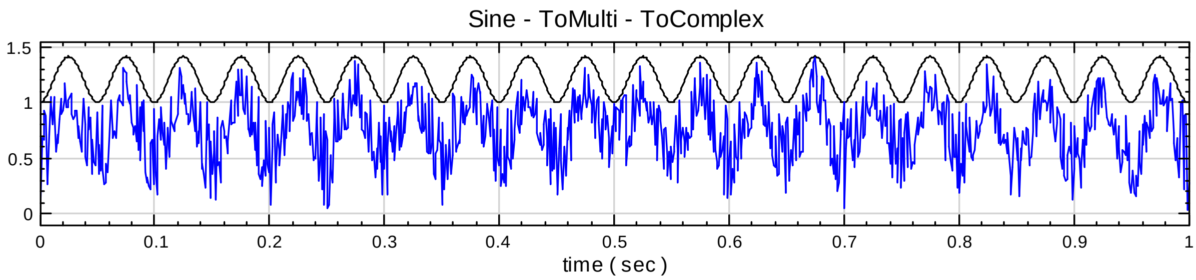
Click on any ToMulti and open the Module in Properties. It shows that the OutputType is Real which means that the signal data are real numbers.



2. Use *Merge to Complex* to convert two multi-channel signals, ToMulti and ToMulti2, to a complex signal. According to the order of channels, this component uses the 1st input signal (Tomulti in this example) as the real part, the 2nd input signal (Tomulti2 in this example) as the imaginary part to make a multi-channel complex signal. In ToComplex, the data of 1st channel is *Sine Wave+ i *Square Wave* and the data of 2nd channel is *Noise+i*Triangle Wave*.

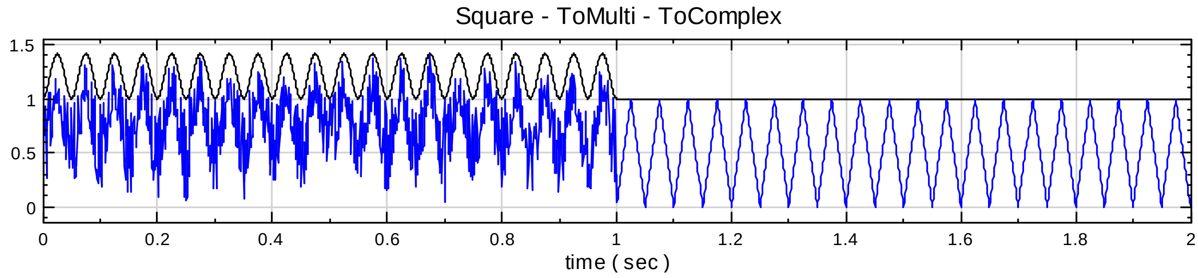


3. Adjust *Reference Input* to decide the time-axis of the ToComplex. It uses the 1st channel as the default. Connect *Channel Viewer* to the ToComplex. Because the *Reference Input* is set to ToMulti, it shows that time-axis length is 1 second and the data between 1 second and 2 second are discarded in ToMulti2.



If the 2nd signal is selected as Reference Input for ToMulti2, the signal diagram will be altered as the figure below. The time-axis length is 2 seconds. Because

there are no data between 1 second and 2 second in ToMulti signal, this part is cleared to be zero automatically. And the imaginary part and Magnitude are preserved.



Related Functions

Noise, Sine, Triangle, Square, Map to Real, Merge to Multi-channel.

4.6 Merge to Multi-channel

This component merges several single-channel signals into a multi-channel signal.

Introduction

Let x_j to be a signal whose time-axis is *j* , y_k to be a signal whose time-axis is *k,* then the merged signals z_m is

$$z_m = [x_m \;,\; y_m \;]$$

where $\begin{cases} m = j, \text{Reference Input} = x \\ m = k, \text{Reference Input} = y \end{cases}$

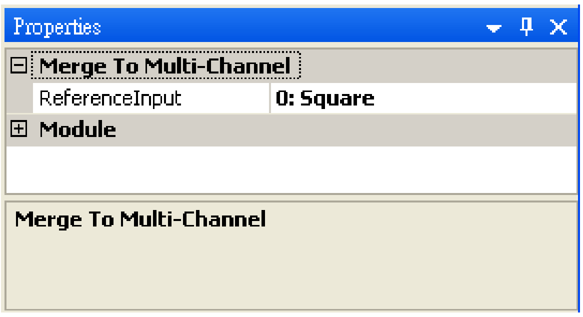
m represents the signal time-axis. If the *Reference Input* is set to *x*, the time-axis of output signal *z* is identical to the one in *x*, i.e. $m = j$, and the time-axis of *y* would be replaced by the time-axis of *x*. Note that this module replaces the coordinates of the time-axis and more attention is needed for the lengths of input signals.

Properties

This module accepts input of Signal (which could be real number or complex number, single channel, Regular or Indexed), Audio (which could be real number or complex number, single channel, Regular), Numeric (which could be real number or complex number, single channel, Regular or Indexed).

In this module, *Reference Input* selects an input signal used as reference of the number of channels and time-axis of the output signal. The default value is 0 which means that the output references the 1st input signal. The time-axis settings of other input signals would be copied directly from the 1st signal. The principle of copying time-axis setting is that the time points of missing data are filled with 0 and time points of exceeding the time reference are discarded.

In order to avoid the operation confusion, it is recommended to use signals with identical settings, such as SamplingFreq, time starting point, and Time Length. Definitions and default values of properties are given below.

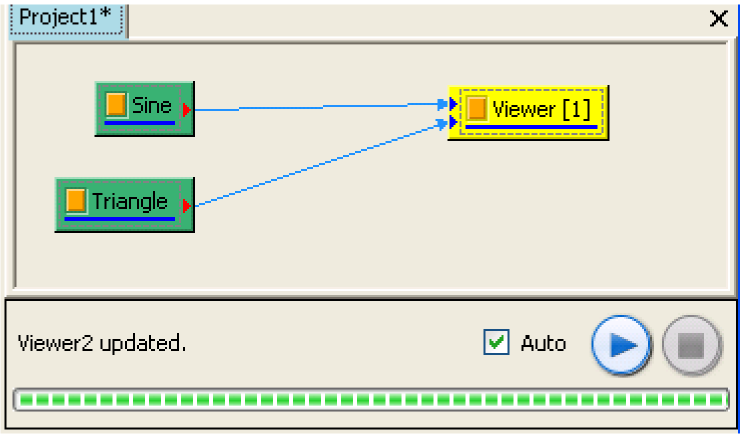


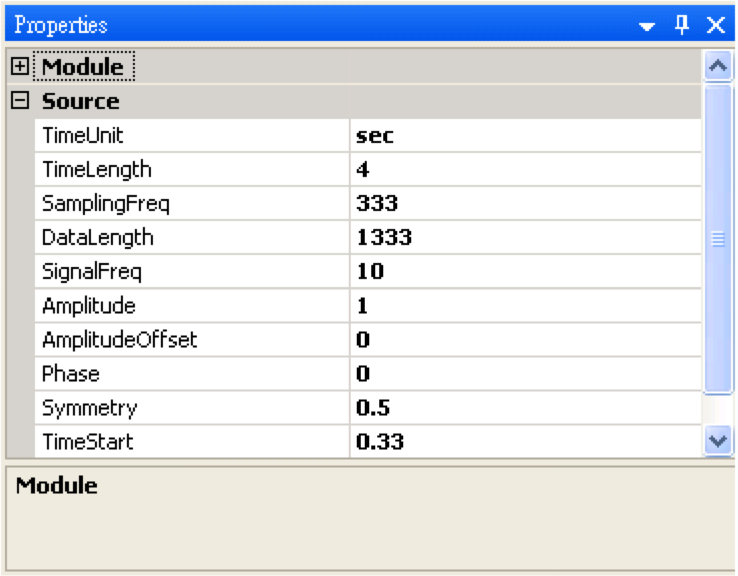
Property Name	Property Definition	Default Value
ReferenceInput	To set the reference signal, its time axis is used as the time axis of the output signal	The 1st input signal

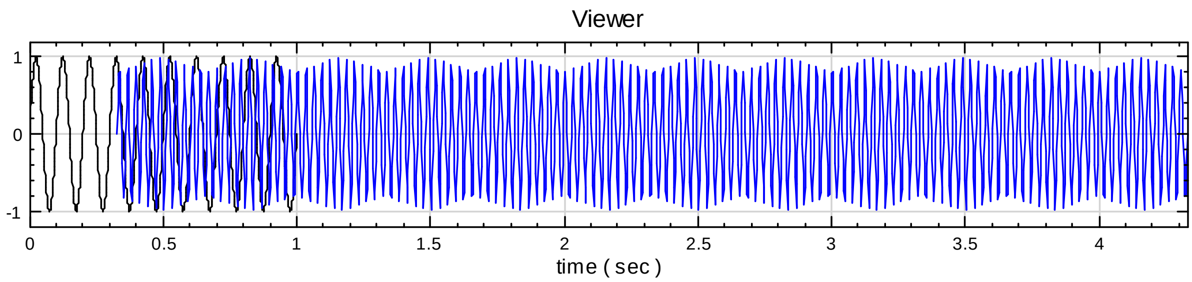
Example

This module accepts Regular, Indexed input signals. The examples show the operation of input signals with identical and different time-axis setting.

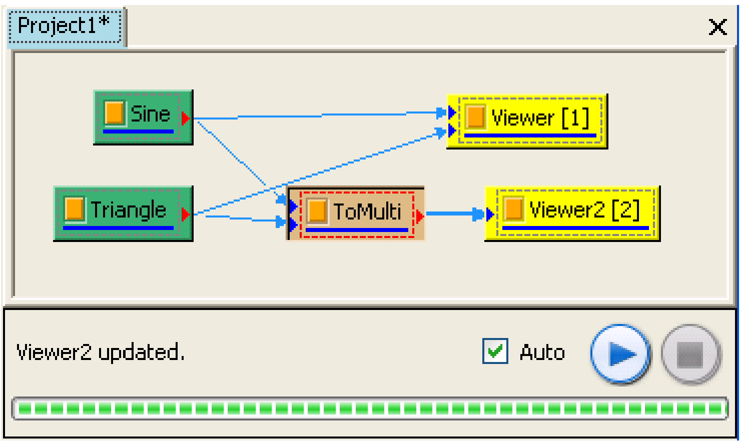
1. Use *Source→Sine Wave* to generate a Sine Wave and use Source→Triangle to generate a triangular signal. Change the *Properties/SamplingFreq* of Triangle to 333, *TimeStart* to 0.33, and *TimeLength* to 4 seconds. Finally, link these two signals to *Viewer→Channel Viewer* to plot figures.

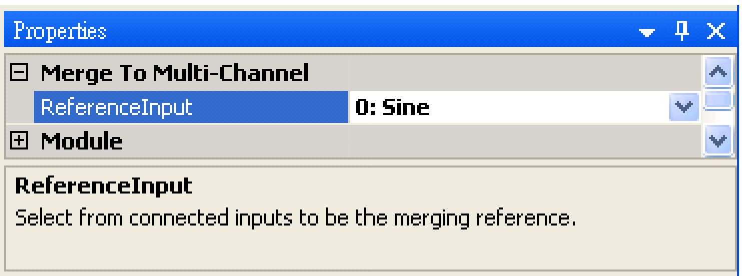


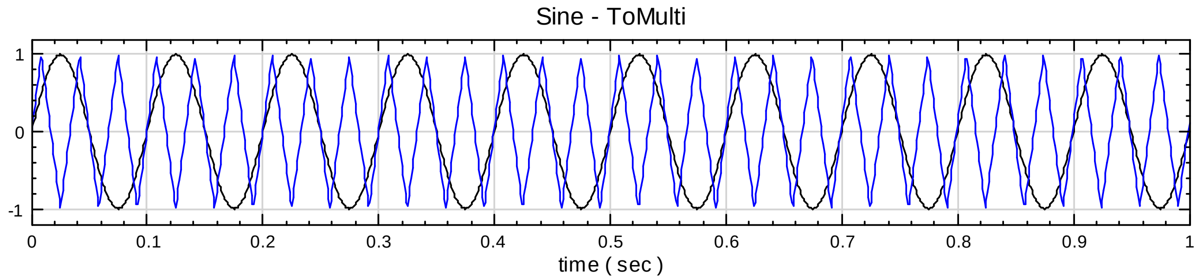




2. Pass these two signals to *Conversion→Merge to Multi-Channel* and use *Channel Viewer* to plot figures.



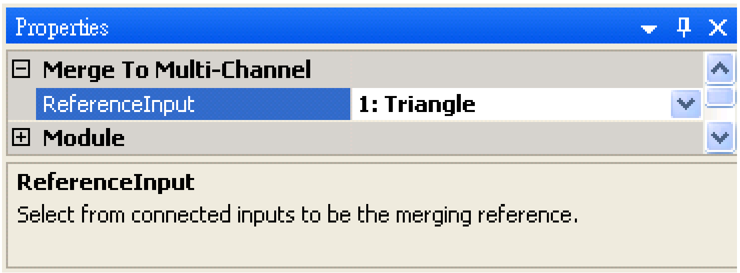


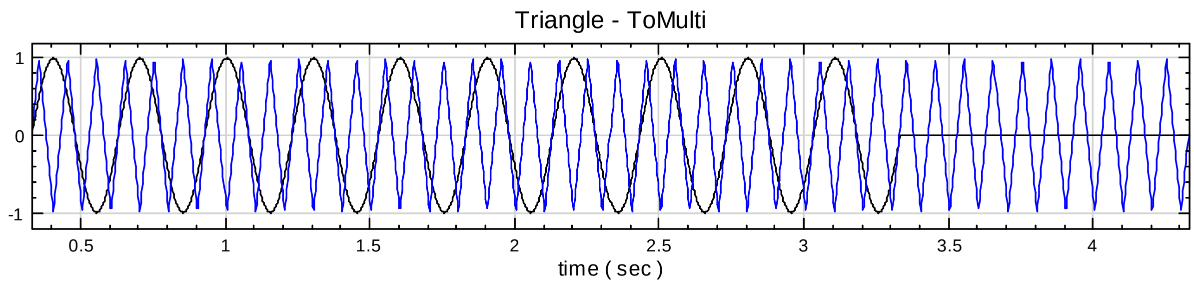


Click on the ToMulti. Because its *Properties/*ReferenceInput is set as Sine, the time-axis setting of the output is identical to those in the Sine Wave, i.e., the time starting point is 0, sampling frequency is 1000Hz, time length is 1 second, and the number of data point is 10001. The contents of Sine are copied to the CH1 in the ToMulti completely.

For the input signal of Triangle, the original time axis would be replaced by the time-axis of Sine. The number of data point in Sine is 1001 while it is 1333 in Triangle. This module would place the first 1001 data points of Triangle to the CH2 in the output signal and delete all other.

3. Change the *Properties/*ReferenceInput to 1:Triangle, the time-axis setting of the output is identical to those in Triangle. Because there are 1333 data points in Triangle while there are only 1001 data points in Sine, the CH1 of the output signal is filled with 0 in the corresponding points which have no data in Sine.

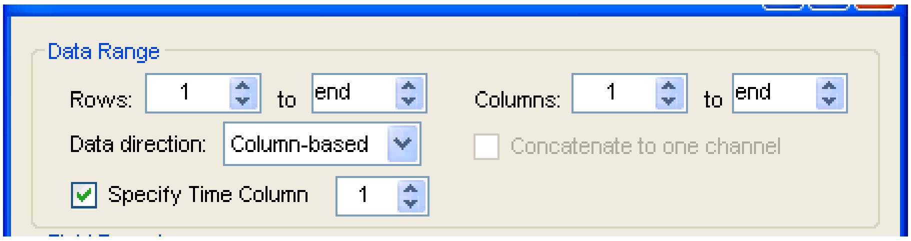




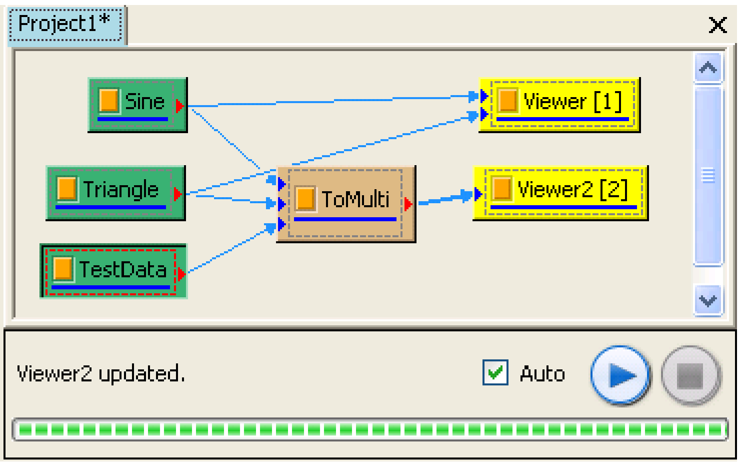
4. Next, read in a set of signals in Indexed format. First, create a simple data set as shown in the figure below, where the 1st column is time and the 2nd column is data.

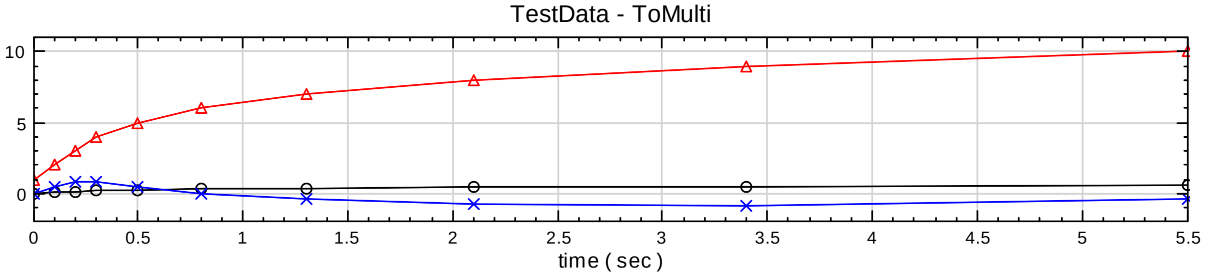
0	1
0.1	2
0.2	3
0.3	4
0.5	5
0.8	6
1.3	7
2.1	8
3.4	9
5.5	10

Next, press in the Network tools or use *Source→Open datafrom file* to read in this data file, TestData.txt. In *Text Importer*, check *Specify Time Column* and then press the confirm button.



5. Not only does Tomulti accept signals in formats of Regular and Indexed, it also accept input signal of mixed Regular and Indexed. Drag TestData to Tomulti and change the *Properties/*ReferenceInput to 2:TestData, the original Regular format in the time-axis of Sine and Triangle signals is replaced by the time-axis of TestData. This is clearer when observing the output of *Channel Viewer*.





Related Functions

Noise, Sine, Channel Viewer.

4.7 Split Complex

This component splits the real and imaginary part of a complex signal or numeric data.

Introduction

Let $Z^{(n)} = \{z_j^{(n)}\}$ be a multi-channel complex signal, where *n* represents channel, *j* represents the time axis, $X^{(n)} = \{x_j^{(n)}\}$ represents the real part, and $Y^{(n)} = \{y_j^{(n)}\}$ represents the imaginary part . Therefore, $z_j^{(n)}$ is denoted as

$$Z^{(n)} = X^{(n)} + iY^{(n)}$$

The output signal of *Split Complex* is

$$S^{(2*n-1)} = \mathrm{Re}(Z^{(n)}) = X^{(n)},\ S^{2*n} = \mathrm{Im}(Z^{(n)}) = Y^{(n)},\ 1 \le n$$

where the odd channels in output signal S save the real part X and the even channels save the imaginary part *Y*. All channels are real data.

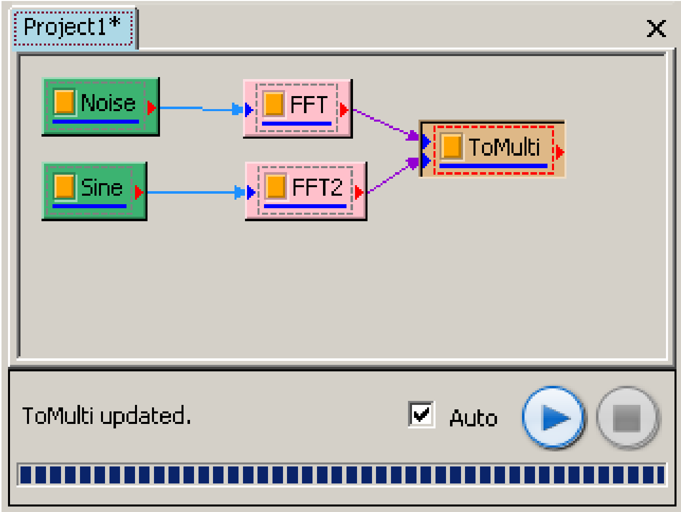
Properties

This module accepts input of Signal (which could be complex number, single channel or multi-channel, Regular or Indexed), Audio (which could be complex number, single channel or multi-channel, Regular), Numeric (which could be complex number, single channel or multi-channel, Regular or Indexed). The output is real, multi-channel data.

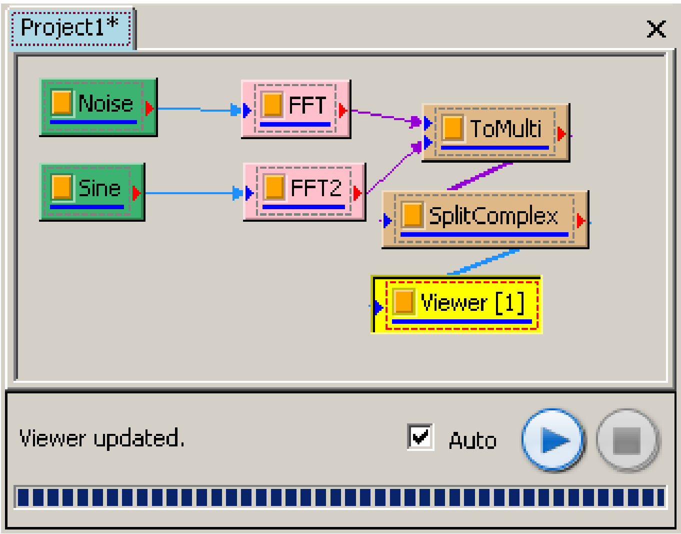
Example

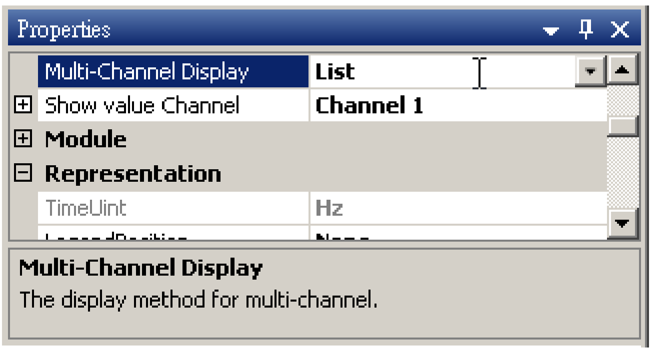
With sampling frequency of 1000Hz, length of 1 second and amplitude of 1, a sine wave and a white noise are chosen as input signals to generate a complex signal by FFT. Then perform *Merge to Multi-channel* and use *Split Complex* to split it into real and imaginary parts.

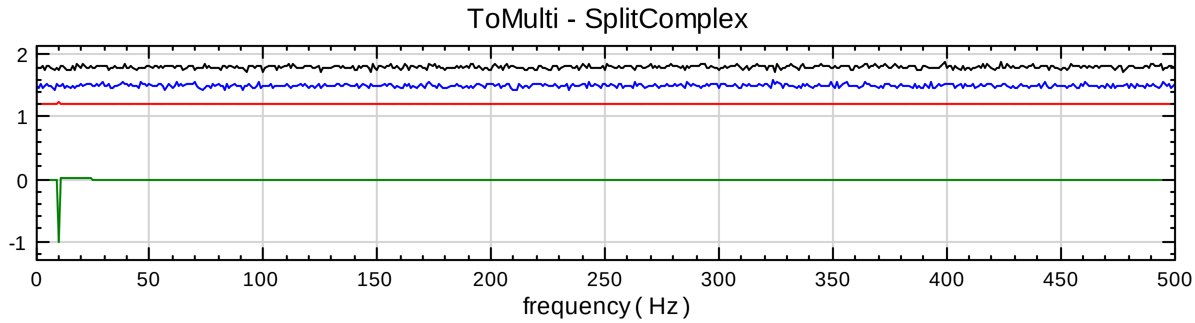
1. In Network Window, use Source→Noise and Source→Sine to generate a white nose and a sine wave. Both signals have the default values in Properties. Use the *Compute→Transfrom→Fourier Transform* to process each signal and then use *Compute→Conversion→Merge to Multi-channel* to merge both signals to form a signal with two channels.



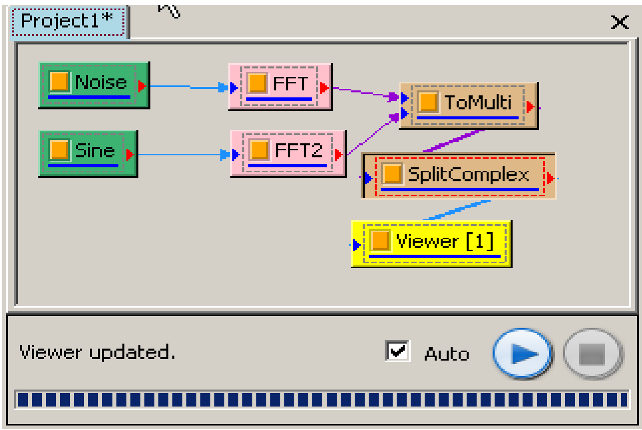
2. Use *Split Complex* to process this complex signal and use *Viewer→Channel Viewer* to show the result. Change *Properties/Channel→Multi-Channel Display* in Viewer to List.

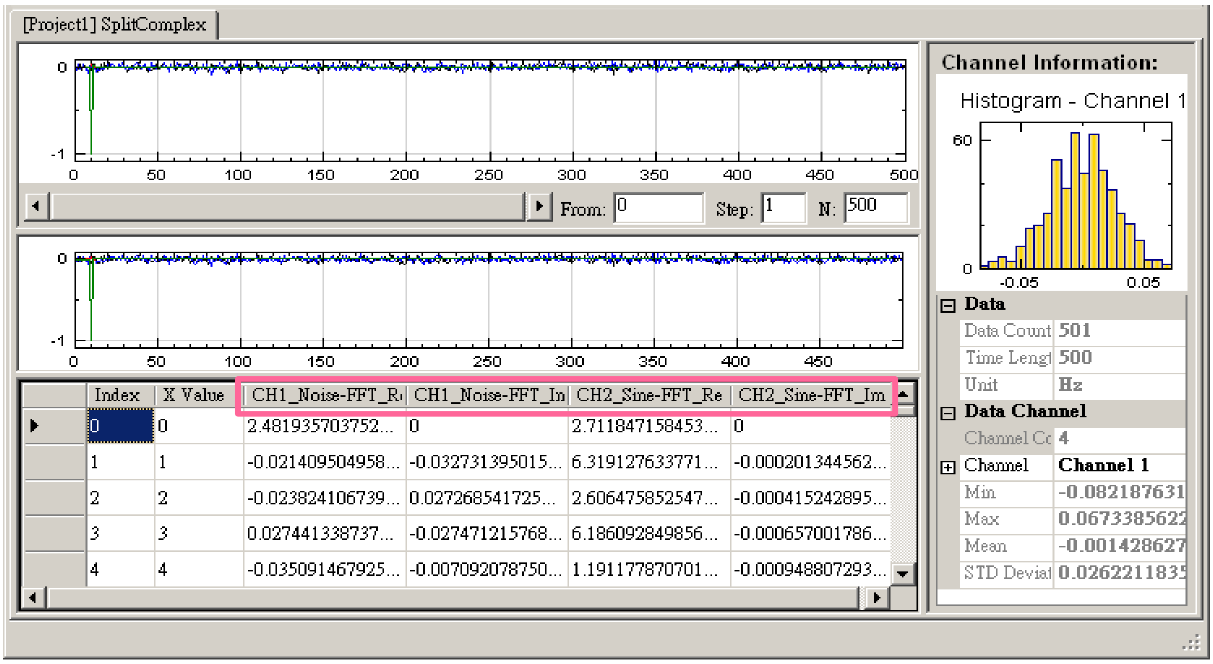






3. Finally, use the *Data Viewer* to observe the output format of *Split Complex.* In the table, the odd channel is the real part of the signal while the even channel is the imaginary part.





Related Functions

Noise, Sine, Merge to Multi-Channel, FFT.

4.8 Convert to Indexed

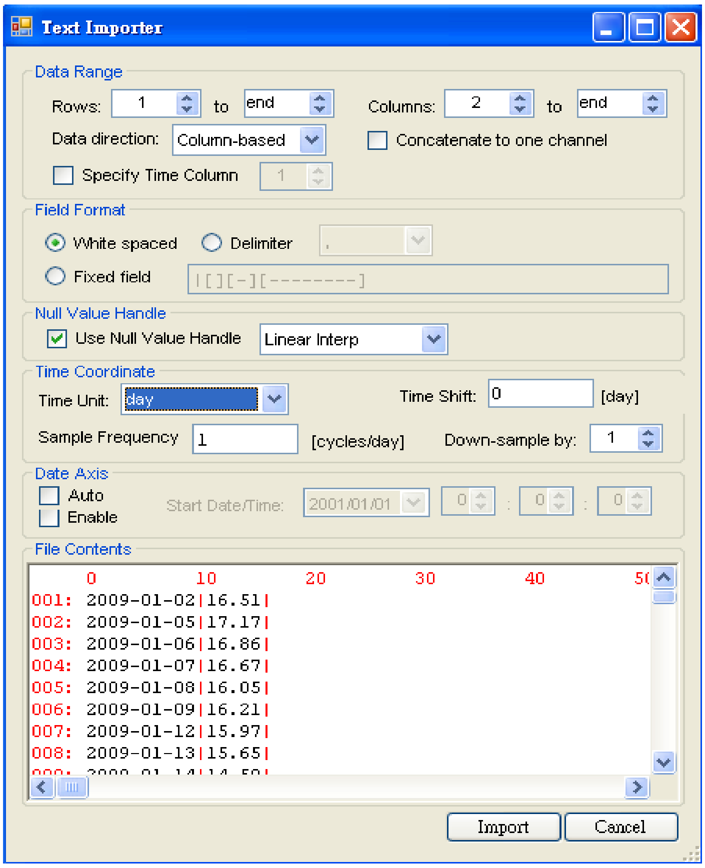
When processing the data, time or date intervals usually are fixed (regular). However, there are exceptions that the time interval of the result is not equal (Indexed). Convert To Indexed can convert a regular data into a new Indexed data based on the uneven time interval of an Indexed signal.

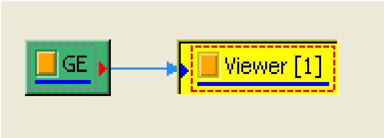
Properties

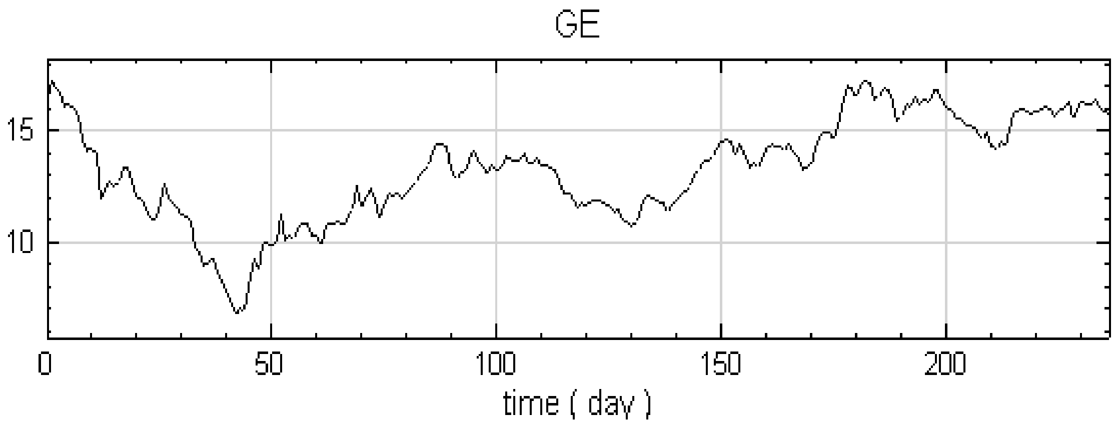
This module accepts two input signals: one Regular signal of Real number, Complex number, single channel or multi-channel, Regular; and another Indexed signal of Regular number, Complex number, single channel or multi-channel, Indexed. The output is an Indexed signal of Real, Complex, single-channel or multi-channel data.

Example

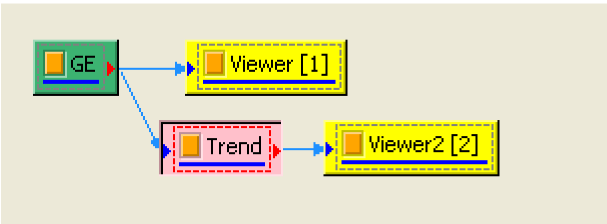
Download GE (General Electrical Co.) stock open price between 2009-01-02 and 2009-12-09 from Yahoo Finance:Yahoo Finance Link and save the data as a CSV file. Since market closes on Saturday, Sunday, and Holidays, we skip to read in the date information for now. Using Text Importer to open the file, uncheck Data Range / Specify Time Column option, set Columns: 2 to end, uncheck Date Axis / Auto option, set Time Coordinate / Time Unit to day, and set Sampling Frequency to 1. So the data read in as Regular and time unit is day. Then connect to Viewer / Channel Viewer for data display.

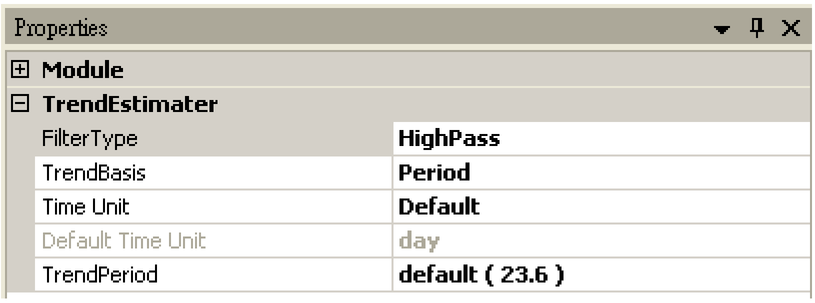


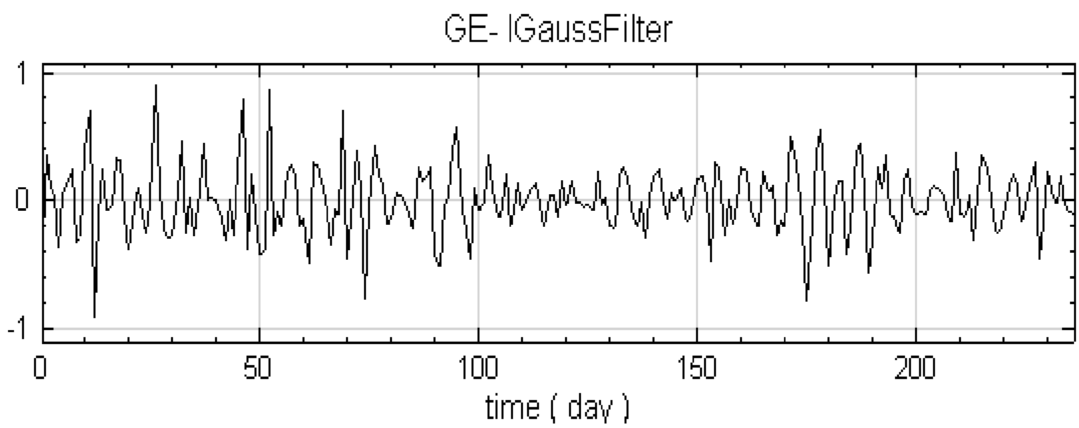




Connect GE Source to Compute / Filter / Trend Estimator. Set Trend Estimator Properties, Filte Type to HighPass. And connect Trend to Viewer /Channel Viewer.

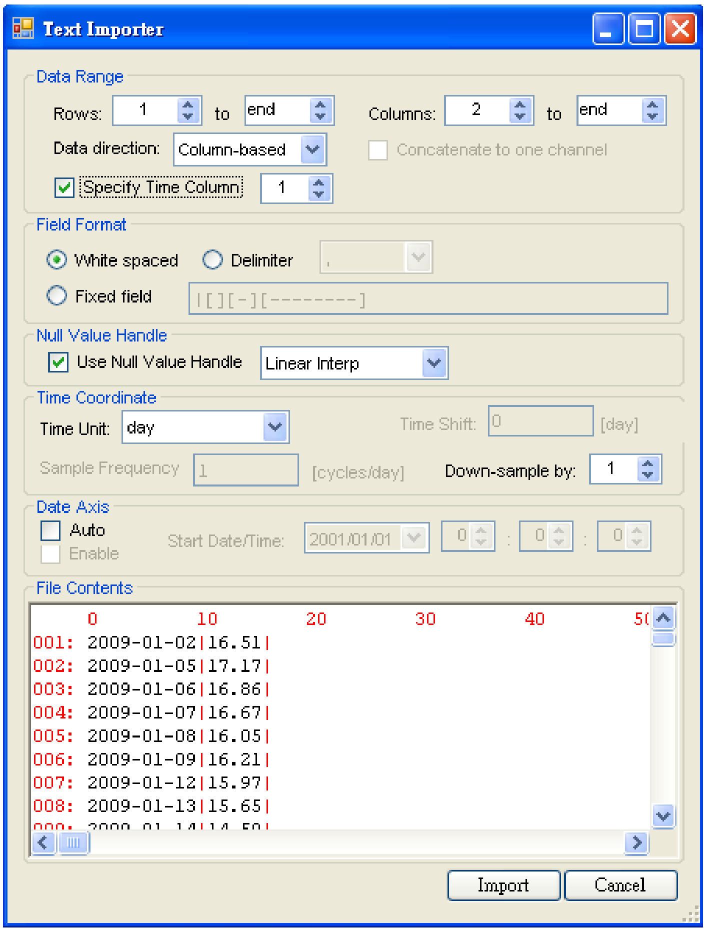




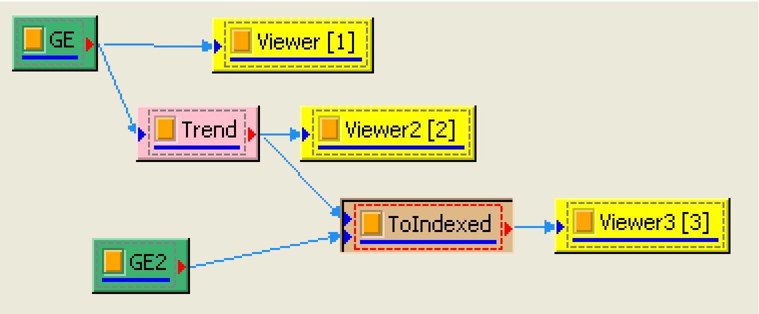


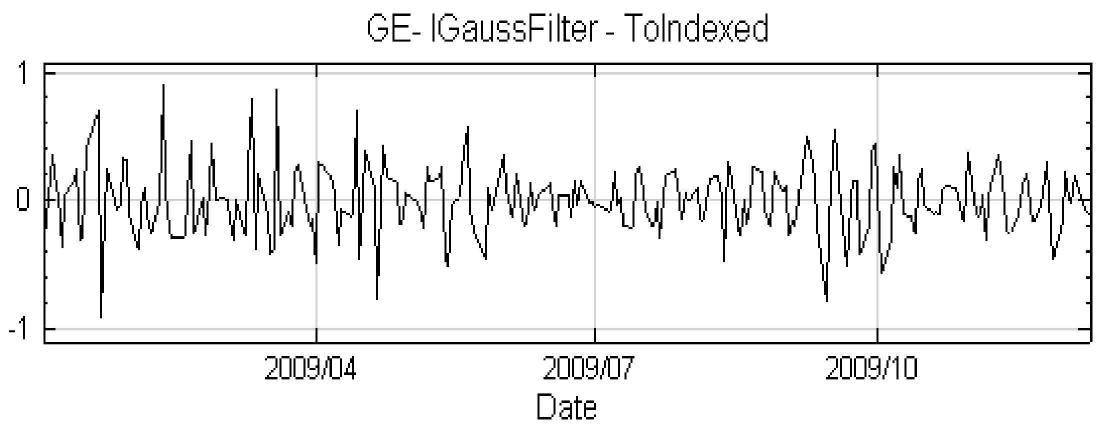
Finally replace the time axis with the date information in the file and convert Regular to Indexed signal. Using Text Importer again to open CSV file, check Data Range /

Specify Time Column option, set Columns: 2 to end. Now the read-in signal is Indexed and time unit is day.

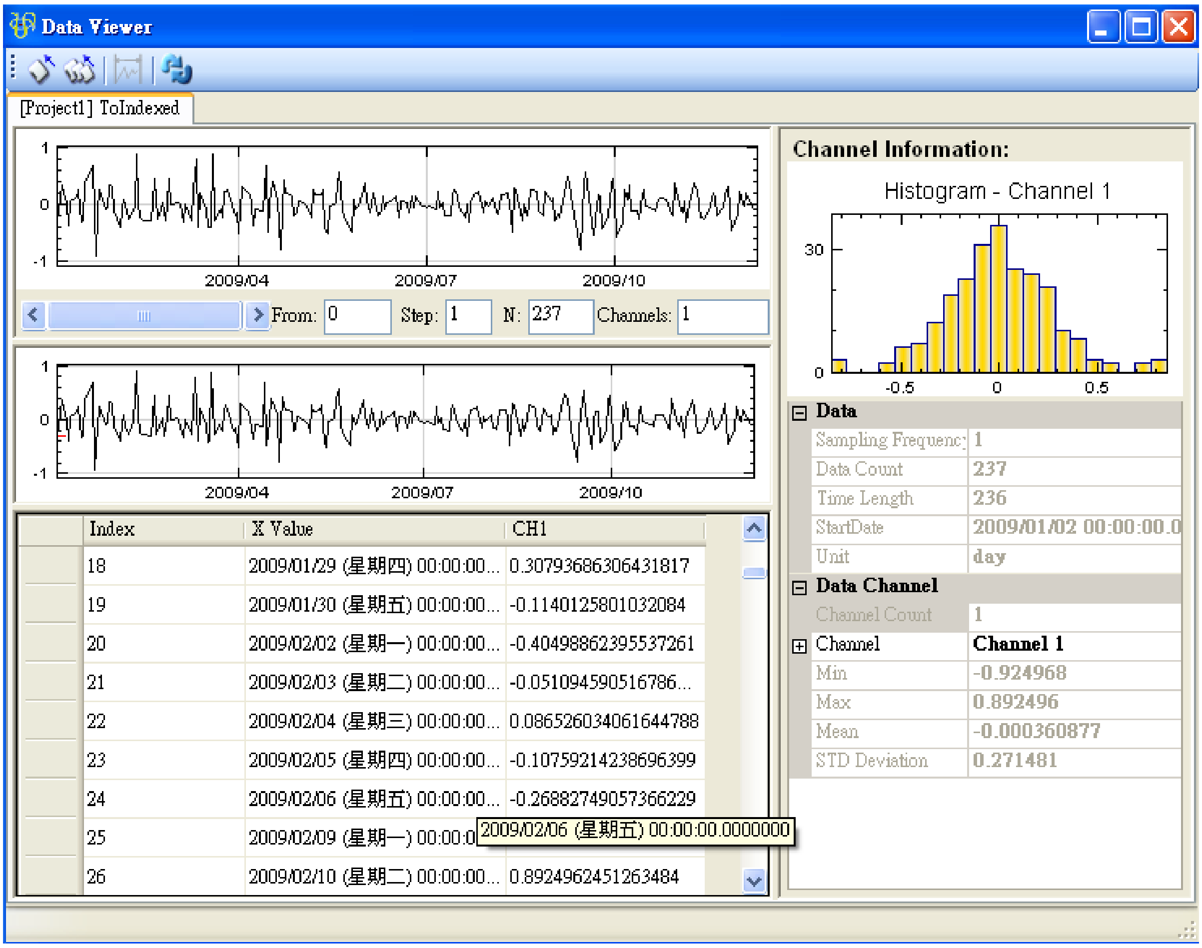


Connect Trend result to Conversion / Convert to Indexed and connect the 2nd Source (GE2) to Convert to Indexed also. Then connect Convert to Indexed to Viewer / Channel Viewer for displaying, the date information is shown on the X axis.





Using Data Viewer can also view Convert to Indexed signal. There is no Saturday, Sunday, and Holidays. And the signal is in Indexed format.



Related Functions

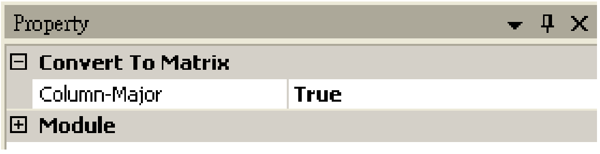
Open Data from File, Channel Viewer, Trend Estimator, Data Viewer.

4.9 Convert to Matrix (Professional Only)

Convert the signal to Numeric matrix and carry out matrix operations.

Properties

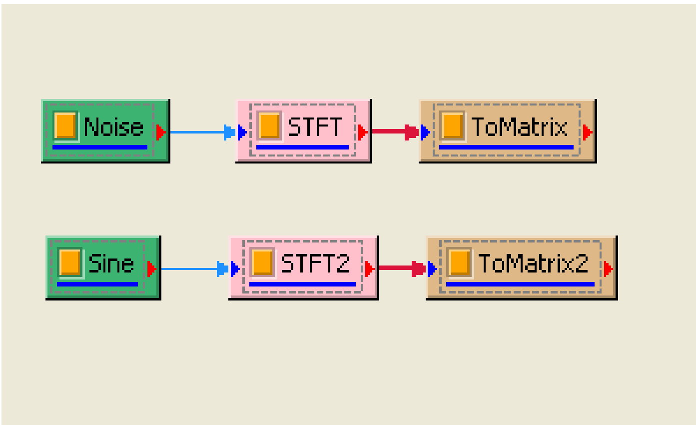
This module accepts input signals of real, complex, single channel or multi-channel, regular, indexed, Audio, and spectra.



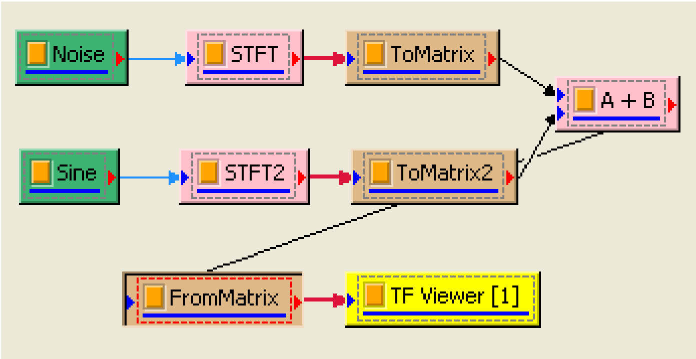
Property Name	Property Definition	Default Value
Column-Major	Time series of input signal is the column of the matrix	True

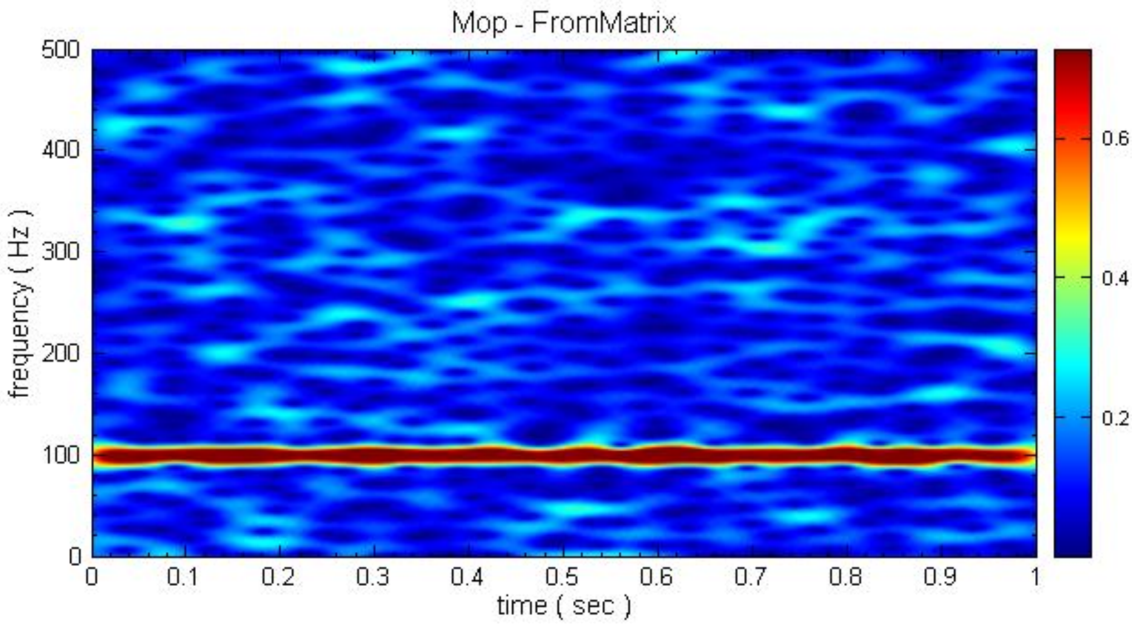
Example

Using Source / Noise and Sine Wave to create two signals, connect them to Compute / TFA / ShortTerm Fourier Transform separately, then convert the signals to matrix with Conversion/ Convert to Matrix as shown below.



Next connect these two matrixes to Compute / Matrix / Matrix Operation with default parameters. Then connect Matrix Operation (A + B) to Coversion / Covert from Matrix, and set DataType to TimeFrequencySpectra, display result with TF Viewer.





Related Functions

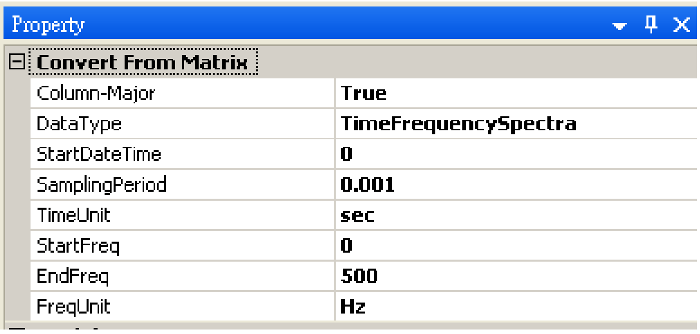
STFT, Convert from Matrix.

4.10 Convert from Matrix (Professional Only)

Convert Numeric matrix to various signals, such as time series, spetra, time-frequency data.

Properties

This module accepts matrix with real, complex, regular, or indexed numeric value.



Property Name	Property Definition	Default Value
Column-Major	Time series of input signal is the column of the matrix	True
DataType	Output signal format: TimeDomainSignal, FreqDomainSignal, TimeFrequencySpectra	TimeDomainSignal

If DataType is TimeDomainSignal, more properties can be set.

Property Name	Property Definition	Default Value
StartDateTime	Start time of output signal	0001/1/1 12:00:00pm
SamplingPeriod	Sample Period of output signal	0.001
TimeUnit	Time unit of output signal	sec

If DataType is FreqDomainSignal, more properties can be set.

Property Name	Property Definition	Default Value
StartFreq	Start frequency of output signal	0
EndFreq	End frequency of output signal	500

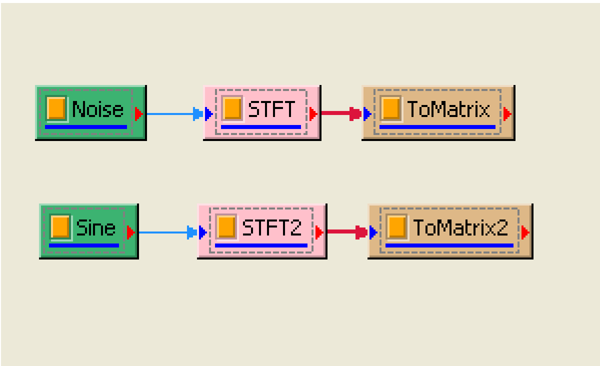
FreqUnit	Frequency unit of output signal	Hz

If DataType is TimeFreqSpectra, more properties can be set.

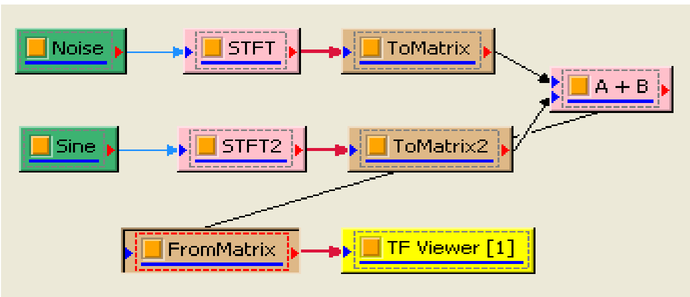
Property Name	Property Definition	Default Value
StartDateTime	Start time of time axis in output signal	0001/1/1 12:00:00pm
SamplingPeriod	Sample period of time axis in output signal	0.001
TimeUnit	Time unit of time axis in output signal	sec
StartFreq	Start frequency of frequency axis in output	0
EndFreq	End frequency of frequency axis in output	500
FreqUnit	Frequency unit of frequency axis in ouput	Hz

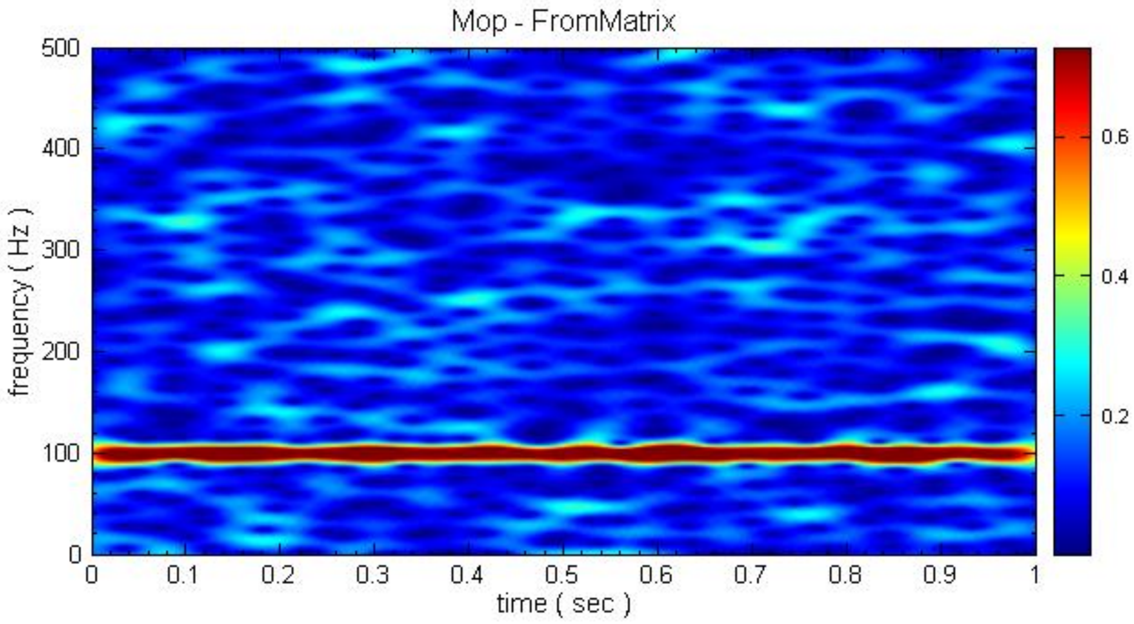
Example

Using Source / Noise and Sine Wave creates two signals, connect them to Compute / TFA / ShortTerm Fourier Transform separately, then convert them to matrix using Conversion / Convert to Matrix as shown below.

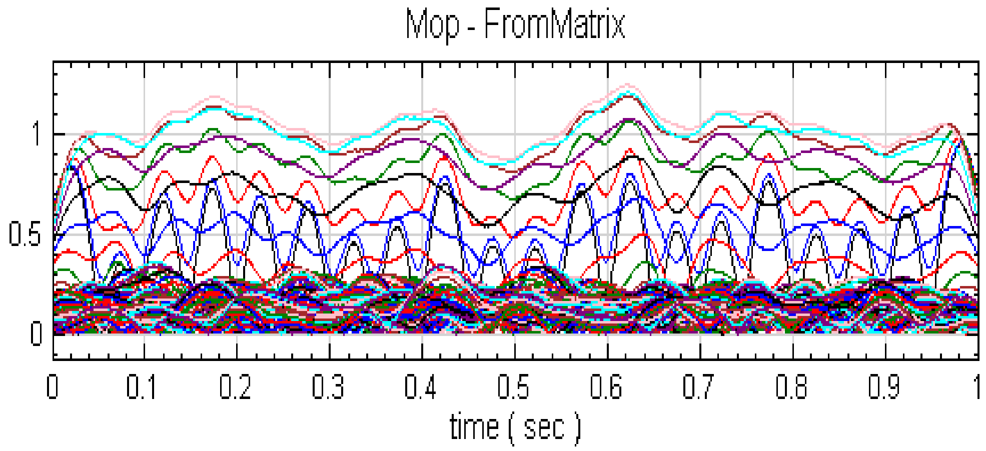


Next connect these two matrixes to Compute / Matrix / Matrix Operation with default parameters. Then connect Matrix Operation (A + B) to Coversion / Covert from Matrix, and set DataType to TimeFrequencySpectra, display result with TF Viewer.





Set DataType to TimeDomainSignal with default parameters, and show the results with Channel Viewer.



Related Functions

STFT, Convert from Matrix.

4.11 Convert from Spectra

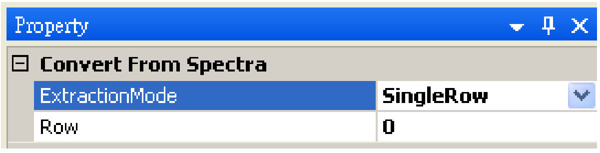
Convert Spectra data to single-/multi-channel time series or single-/multi-channel frequency distribution signal.

Introduction

We can extract one row or all rows from the Spectra, which is the amplitude time series at a fixed frequency. We can also extract one column or all columns from the Spectra, which is the frequency distribution at a fixed time point.

Properties

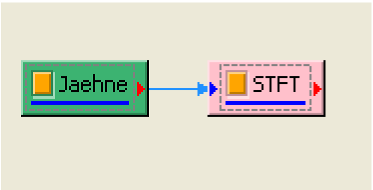
This module accepts spectra with real, complex, single channel, or multi-channel data.



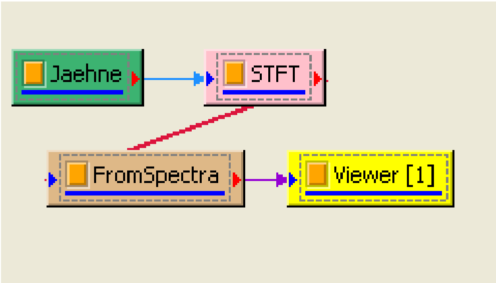
Property Name	Property Definition	Default Value
ExtractionMode	Extract Row or Column, options: MultiChannelRows, MultiChannelColumns, SingleRow, singleColumn	SingleRow
Row	Set which row to extract	0
Column	Set which column to extract	0

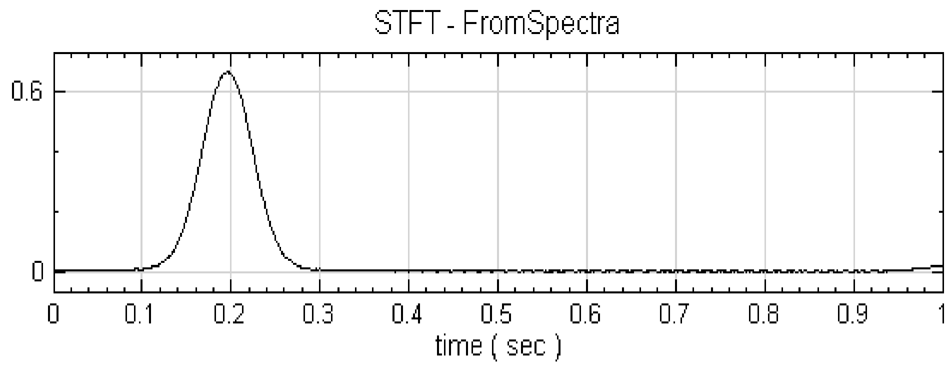
Example

Using Source / Advanced / Jaehne to create input signal and connect the signal to Compute / TFA / ShortTerm Fourier Transform. All use default settings.

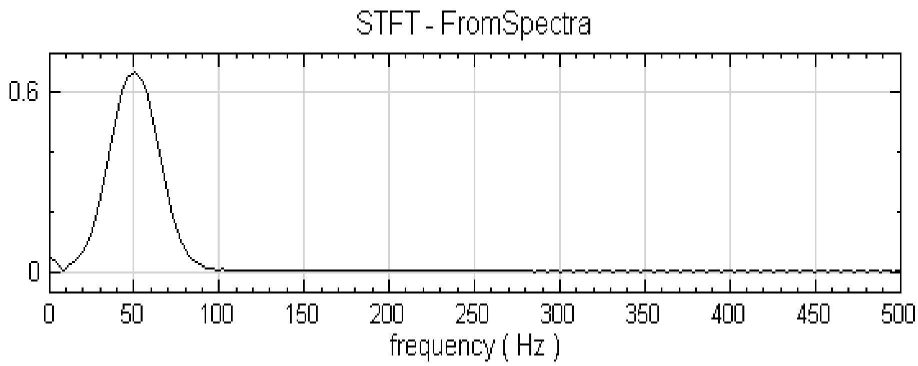


Connect STFT to Coversion/ Convert from Spectra and set Rows to 50 in Convert from Spectra properties, then display 50th row data using Channel Viewer.





In Convert from Spectra properties, set ExtractionMode to SingleColumn and set Column to 100. Show the frequency distribution of the 100th column using Channel Viewer.



Related Functions

STFT, Convert from Matrix.

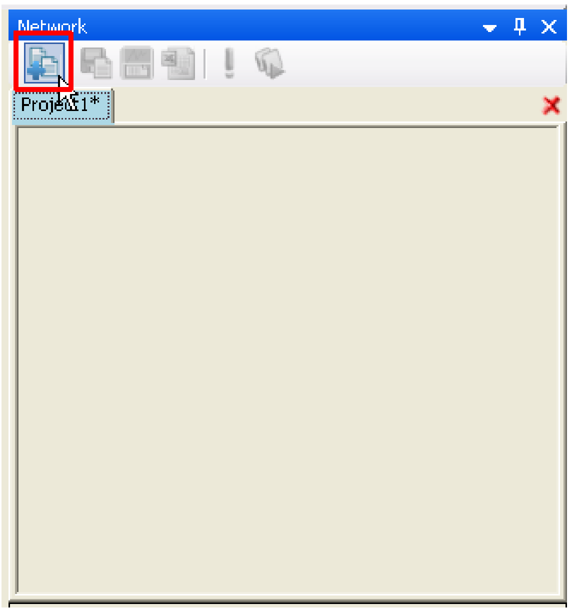
Chapter 5: Source Signal Flow Object

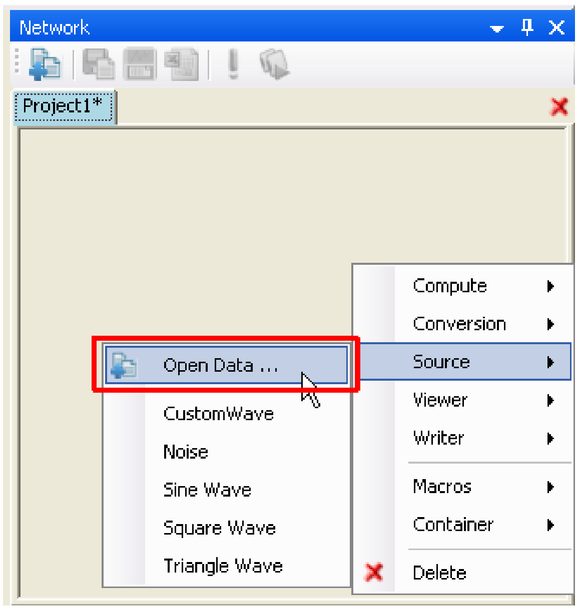
5.1 Open Data

Open a file to be used by DataDemon.

Properties

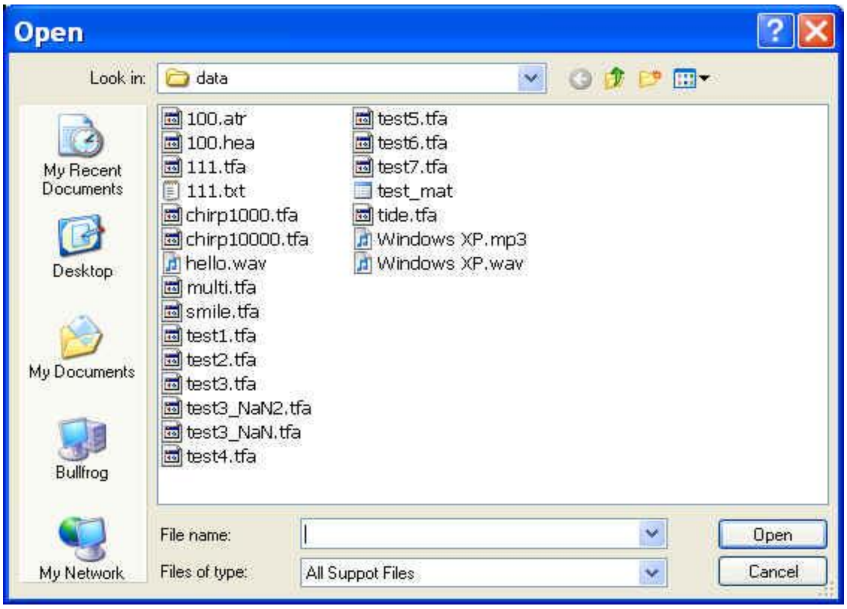
There are two methods to open a data file. The first method is to click on the *Import data from file* button and select the file you want to load into DataDemon. The second method is to right mouse click the *Network Workspace* and the *Network Workspace Menu* will pop up. From the menu, select *Source*→*Open Data* to select a file to be loaded into DataDemon.

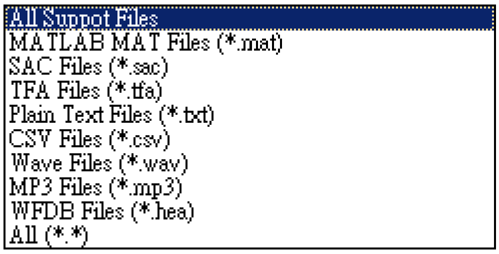


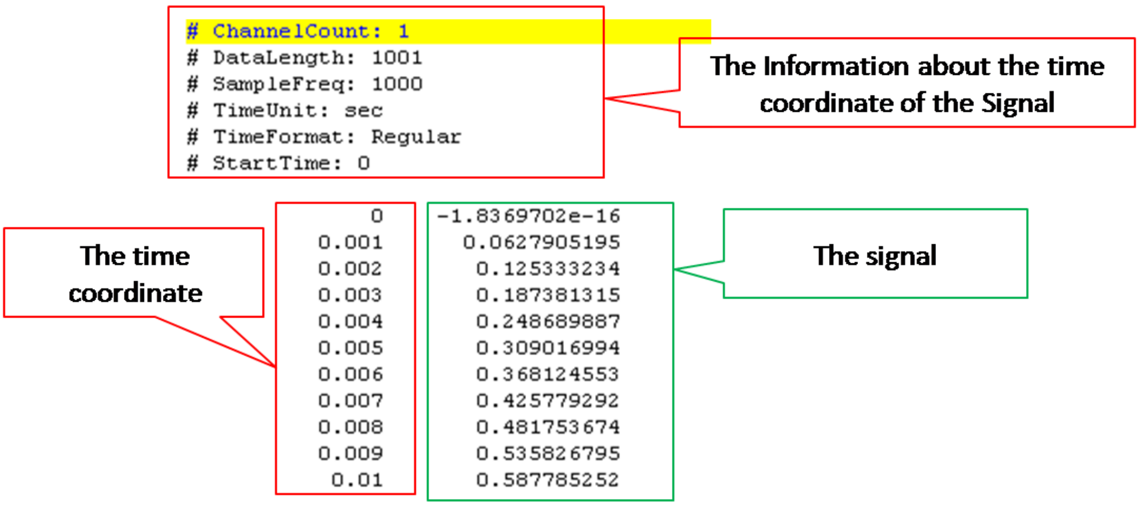


Supported File Types

Using either of above two methods, a browser window is opened for displaying all supported files. If you click on the file type drop down menu, a list of supported file types will be shown. The supported file types includes: .txt, .csv, .tfa, .wave, .mp3 and special file types like MATLAB MAT, SAC for seismology, HEA for MIT WFDB, and .tfa for DataDemon. When you open a .tfa file, the lines which begin with "#" contain information about the detailed aspects of the data and other lines are data signals. So a .tfa file not only has signals, but also has meta data information.



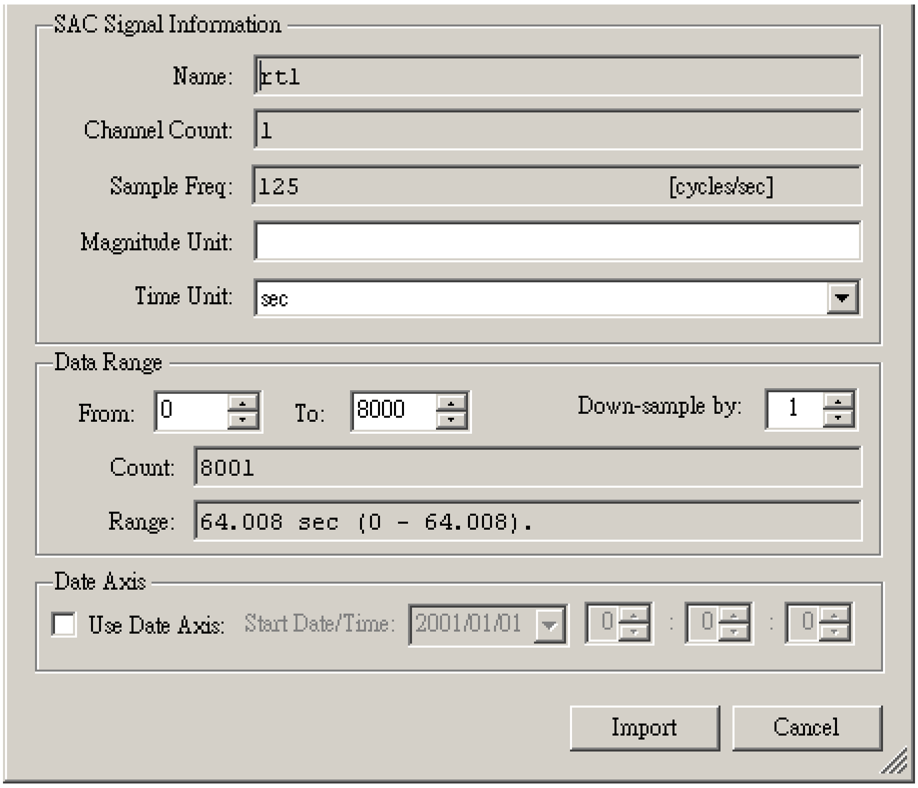




When opening a SAC file, the SAC Importer will pop up. When opening a HEA file, the WFDB Importer will pop up and when opening any other type of files such as .csv and .txt, the Text Importer will pop up.

5.1.1 SAC Importer

SAC file type is used for seismology and opening a sac file will open up a SAC Importer.



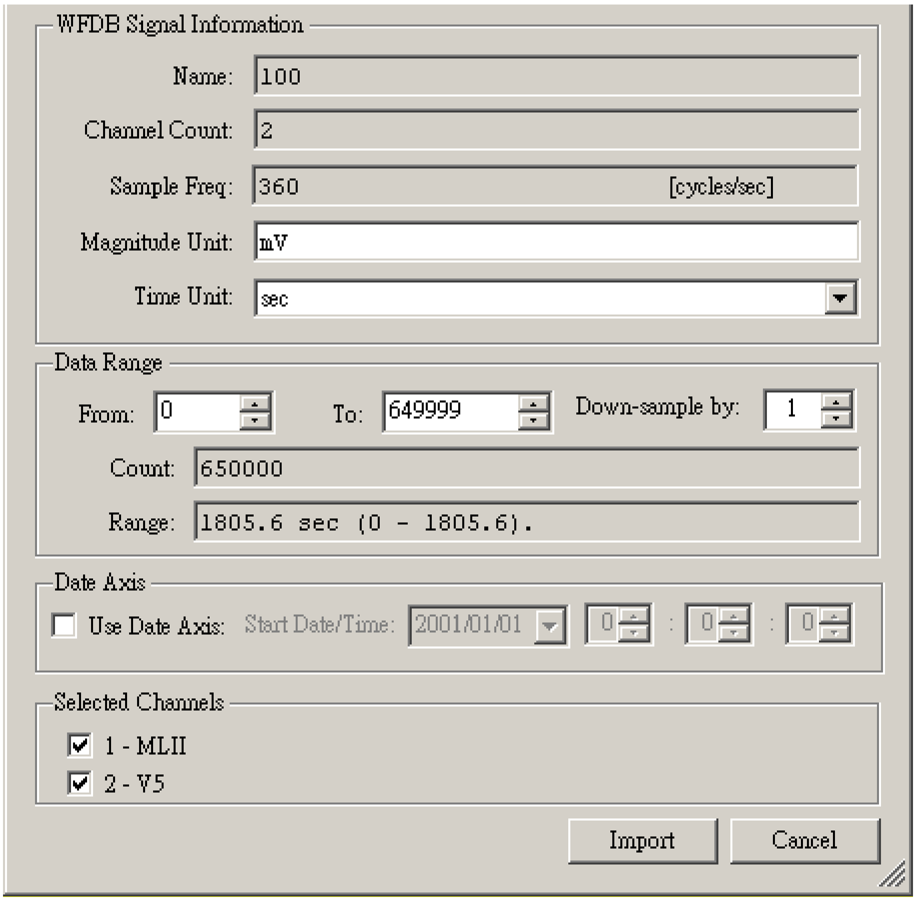
SAC Signal Information displays the *Name*, *Channel Count*, and *Sample Freq* of the loaded file. User can configure the *Magnitude Unit*, *Time Unit* and the *Data Range* of the signal data.

Data Range allows the user to select a signal range by entering the desired value in the *From field* (start position) and *To field* (end position). User can also select the unit for *Down-sample* if the user wishes to down-sample the data.

The user can check *Data Axis* to add a date and a time to the time information.

5.1.2 WaveForm Database Importer

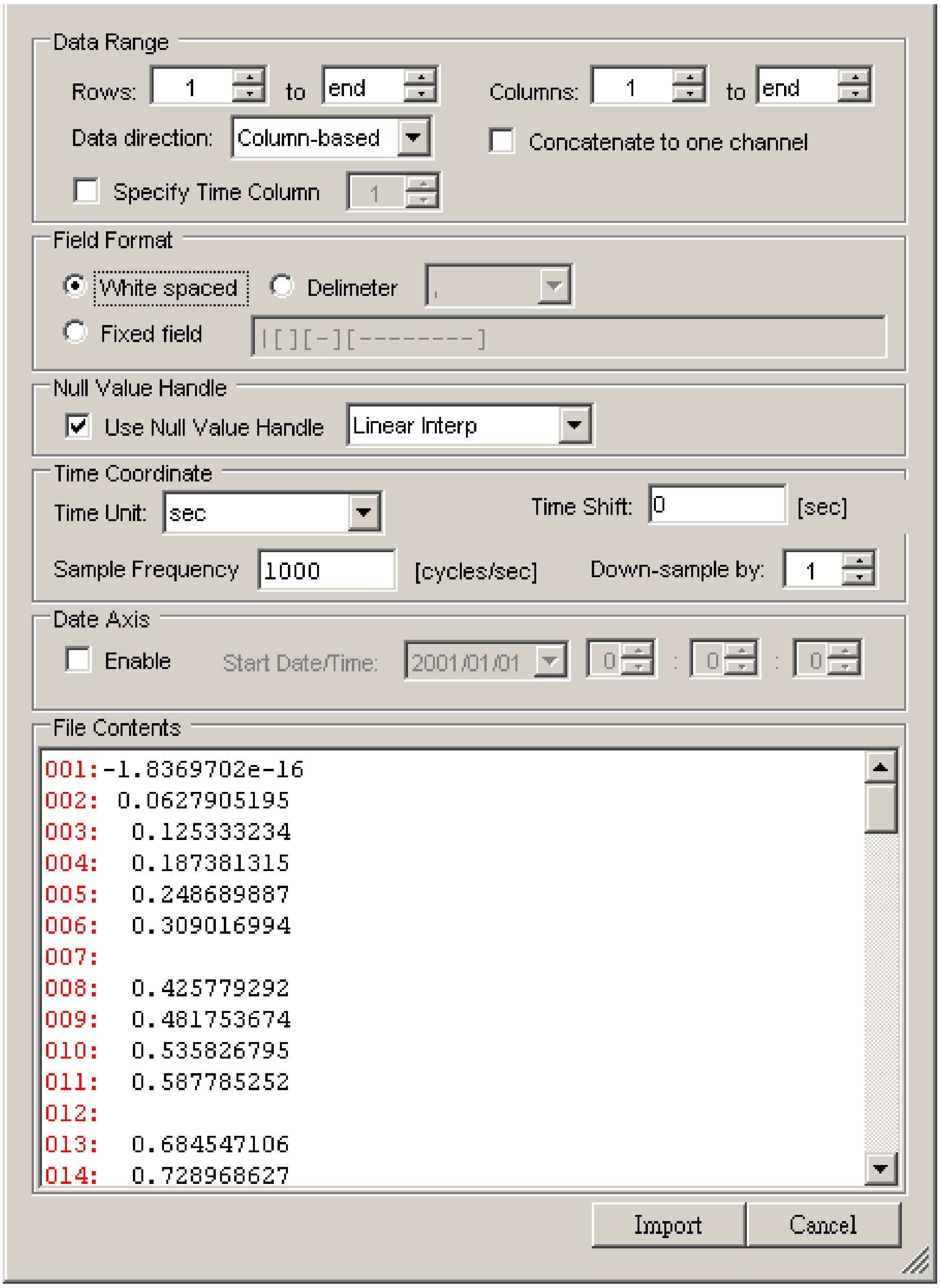
HEA file type is MIT WFDB format. It is used for physiological signals. Opening a HEA file opens up a WFDB Importer.



WFDB Importer and SAC Importer are relatively similar. They both have *Signal Information*, *Data Range* and *Data Axis*. The only difference is that WFDB has an additional option of the *Selected Channels*.

5.1.3 Text Importer

Opening .csv and .txt file types will open up a Text Importer. The Text Importer is more complicated than the other two importers.



The fields in Data Range are explained in the table below.

Property Name	Property Definition	Default Value
Rows	Enter the range of rows to be read.	1 to End
Columns	Enter the range of columns to be read.	1 to End
Data Direction	Determine the way to read the data, either row based or column based.	Column-based
Concatenate to one channel	Determine if the data is to be displayed in one channel or multiple channels (uncheck)	Unchecked
Specify Time Column	Determines if the time information already exist in the signal data. Check to select the column representing the time information. **NOTE:** After checking the box the data will be displayed in the Index format.	Unchecked

In Field Format there are three options to select: White Spaced, Delimiter and Fixed Field. User can select the White Spaced option to separate each data by the white spaced character (most .txt files go by this method). User can select the Delimiter option and choose from the drop down menu to separate each data either by "," character or the TAB character. User can select the Fixed Field option to customize how the data is to be read. The "|" character allows one character to be read from each row to form a channel, "[ ]" character allows two characters to be read from each row to form a channel and "[-]" character allows three characters to be read from each row to form a channel. To read more than three characters into a channel, just add one "-" into "[-]" which becomes "[--]" to read four characters into a channel.

Example:

If " | [--] [] " was entered in the Fixed field to read from a row with the numbers "123456789", then "1" is included in the first channel, "2345" is included in the second channel, "67" is included in the third channel, "8" and "9" are disregarded.

NULL Value Handle option allows user to choose a method to fill in missing values such as *NULL* or *NaN*. Currently methods are *Fixed value*, *Prev value*, *Next value*, *Linear Interp*, *Spline Interp* and *Monotonic Cube*. (**TIPS:** For more information

on filling in missing values, please look up on Resample in Chapter 3.1.7).

Property Name	Property Definition	Default Values
Time Unit	Select the time unit from psec, nsec, msec, sec, minute, hour, day, week, month (30days) and year (365 days).	sec
Time Shift	Set the starting time of the data.	0
Sample Frequency	Set the Sample Frequency.	1000
Down-sample by	Set the Down-Sample rate. With every increment of the value, the sample data is reduced to save time during calculation. (**Note:** The Sampling Frequency value will be automatically recalculated depending on the down-sample value. E.g. Sampling Frequency=1000 with Down-sample=2 will result in creating an imported Source SFO wtih Sampling Frequency=500).	1

Examples

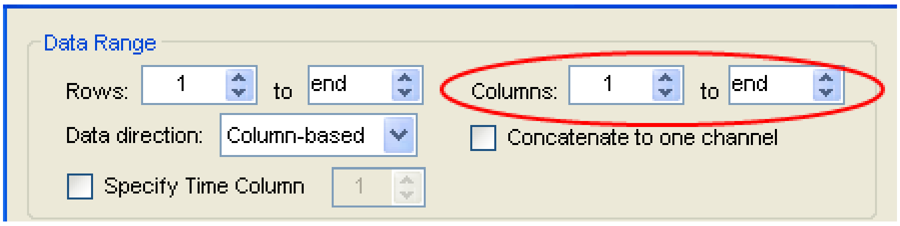
1. Import a Multi-Channel Signal Data.

Load a multi-channel signal data file. The data is separated by white space character and there are three groups of data (one column is one channel)

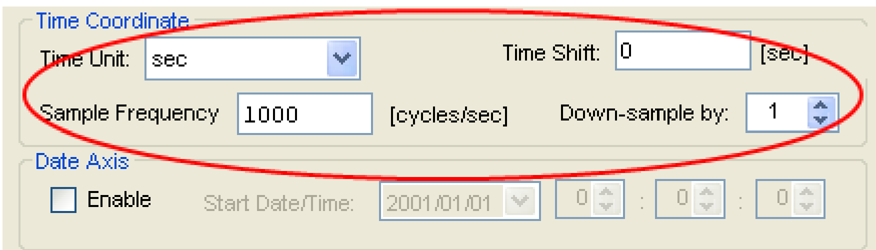
3.28979	-2.69165	-2.03369
3.22998	-2.81128	-2.09351
3.22998	-2.63184	-1.97388
3.34961	-2.75146	-1.91406
3.28979	-2.69165	-1.97388
3.58887	-2.57202	-1.85425
3.46924	-2.81128	-1.97388
3.22998	-2.57202	-2.03369
3.40942	-2.45239	-1.85425
3.17017	-2.75146	-1.85425
3.22998	-2.81128	-2.09351
3.40942	-2.75146	-2.09351
3.46924	-2.69165	-1.97388
3.34961	-2.63184	-2.03369

1. Click on the *Import data from file* button in the *Netowork Window Toolbar* or open it from right mouse clicking on the *Network Workspace* to open up the *Network Workspace Menu* and select *Source*→*Open Data*.

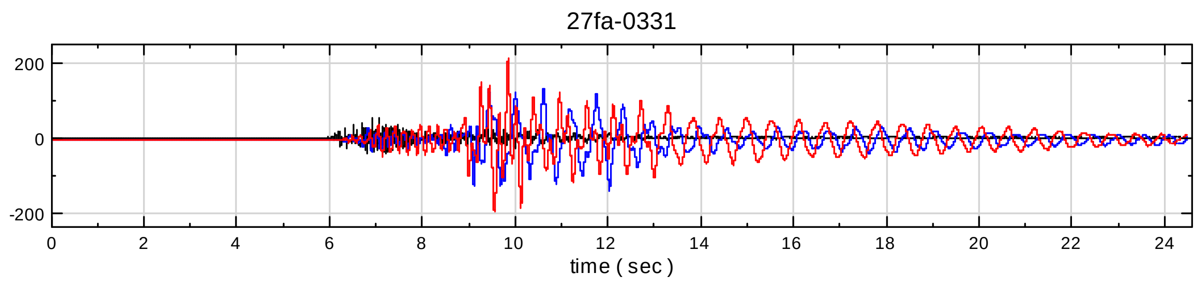
   The Text Importer will pop up when you have selected the text file to import (lets assume the multi-channel text file was selected as describe above). If you want to import all three data columns then leave the Column option as 1~end. (**TIPS:** Set the Column option as 2~2, if you only want to import the second data column).



2. Because the imported data does not contain any time information, the *Time Unit* will be set to *seconds* and *Sample Frequency* set as 1000.

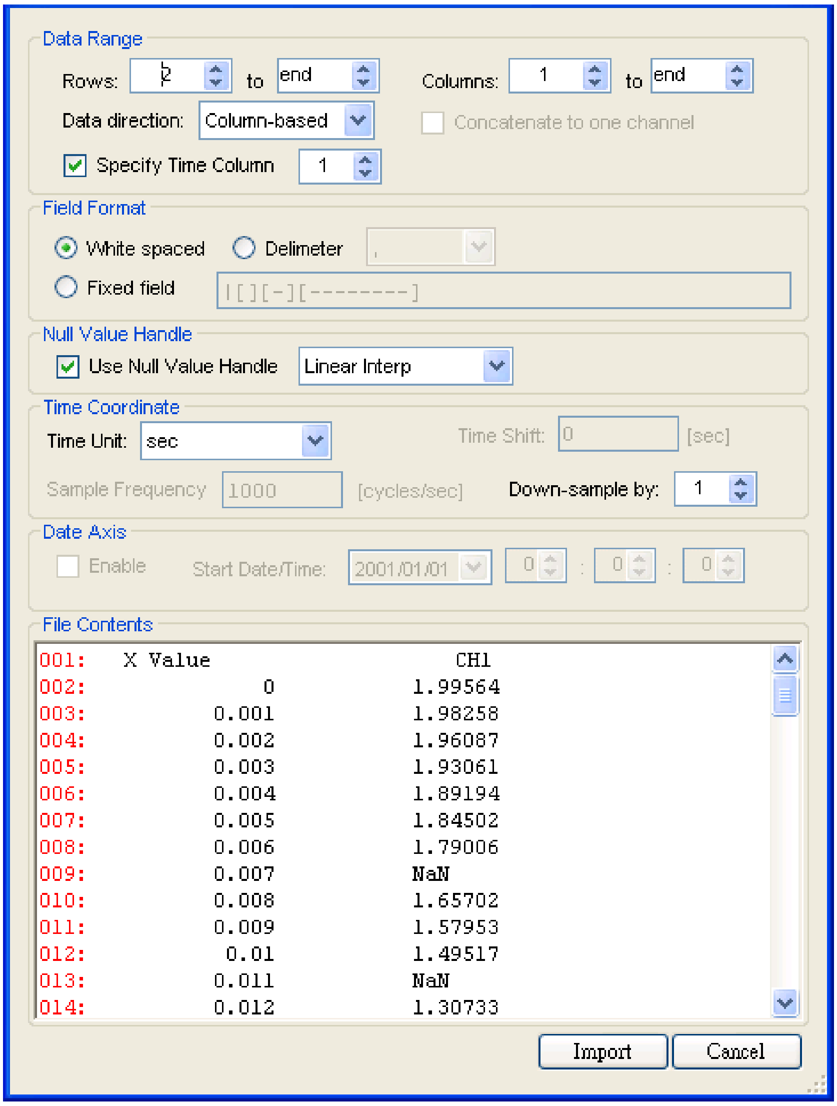


   Click on the Import button to import the data.



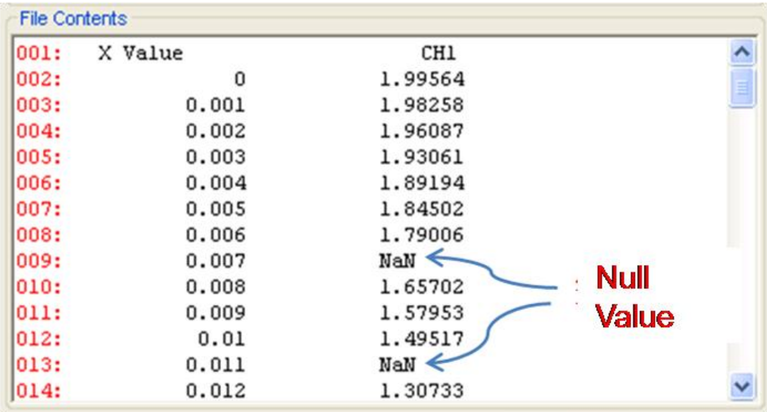
2. Import a data file which has time information and some missing data values.

This example demonstrates how to import a data file which has time information included but contains some missing data values.

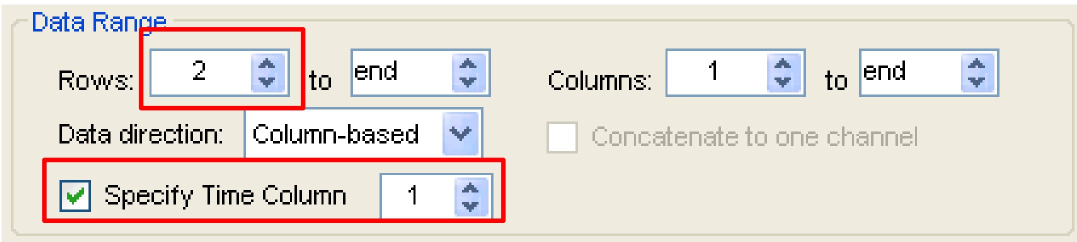


1. First the data to be imported has to be understood. You can see that there is *NaN* (missing data value) in 009 and 013 of the CH1 column. The first column is

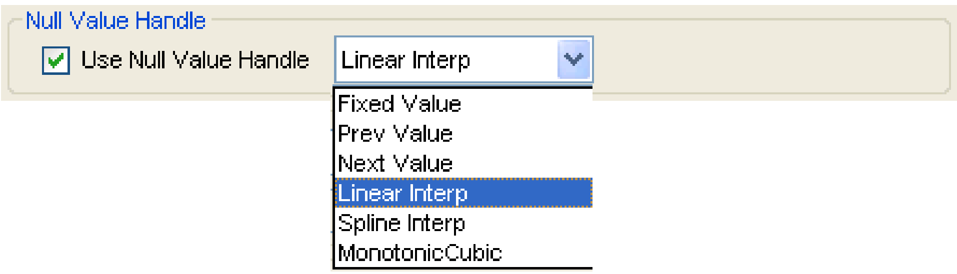
the x value (time) and the second column is the CH1 data value. Text Importer will be configured to import this data properly into DataDemon.



2. The first row in the data contains the titles for the two columns. So in the Rows option under *Data Range*, the Rows should begin from 2 (the second row is where the data values begin). Because the data contains time information, check the *Specify Time Column* option under *Data Range* and select 1 (first column of the data is the time information).



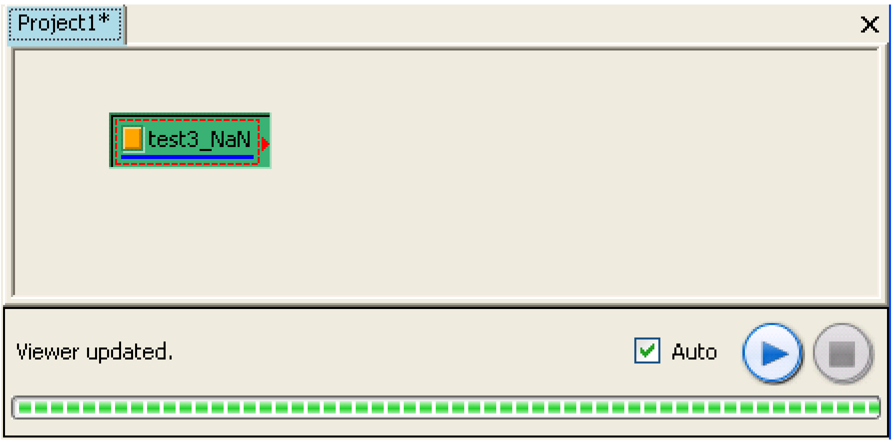
3. The *Field Format* does not need to be edited because the data is separated by *White Space* (which is the default selection). Check *Use NULL Value Handle* option under *NULL Value Handle* and select *Linear Interp* calculation from the drop down menu to fill in the *NaN* values (missing value).



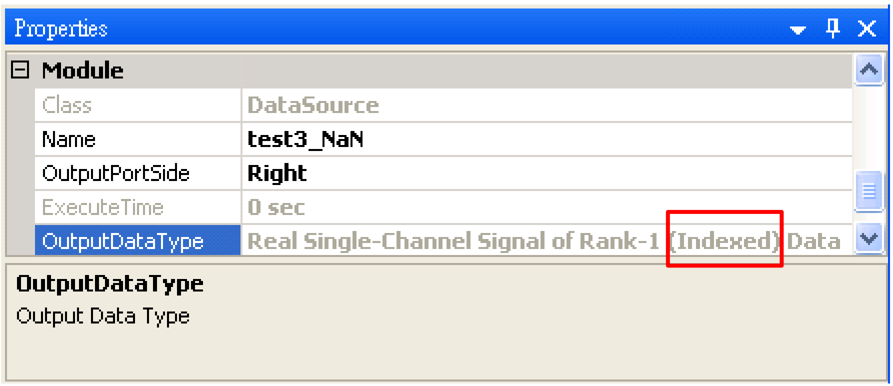
If there are *NULL* or *NaN* values in the data but *Use NULL Value Handle* option isn't checked, then the following warning message will appear.

4. Although the time information exists in the data, you still need to set the unit of the time information. Click on the *Time Unit* option under *Time Coordinate* and select *sec* from the drop down menu.

Click on the Import button once the configurations are done.

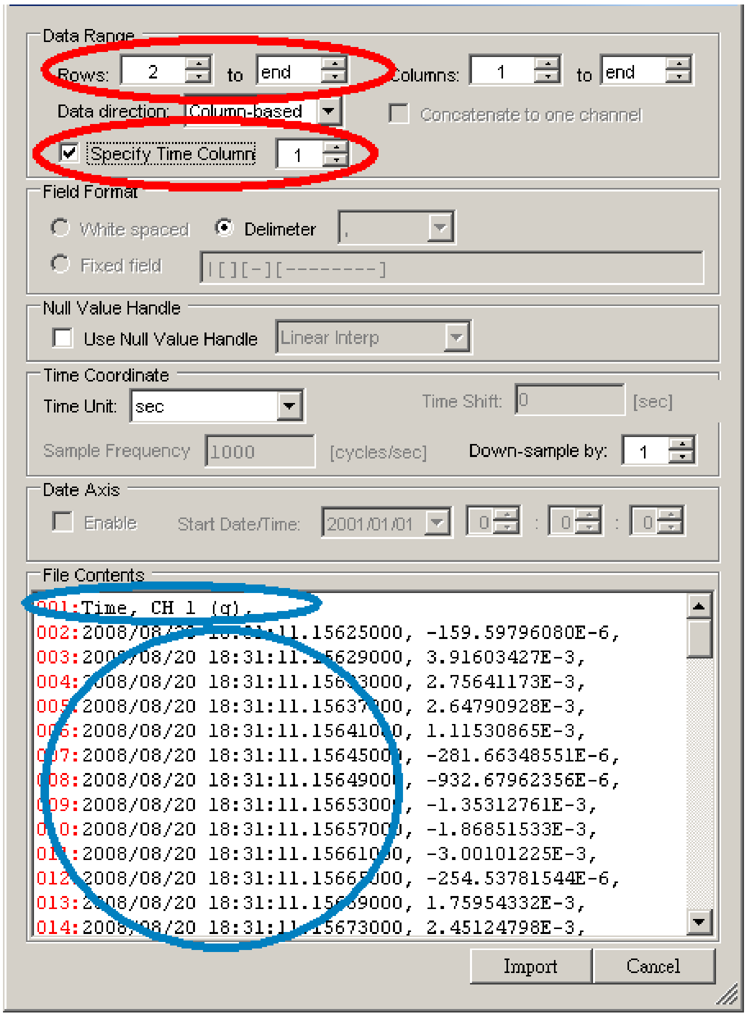


If the signal data is to be calculated further, it needs to connect to *Conversion→ Convert To Regular* because *Specify Time Column* is in the Indexed format. It needs to convert it to regular format.

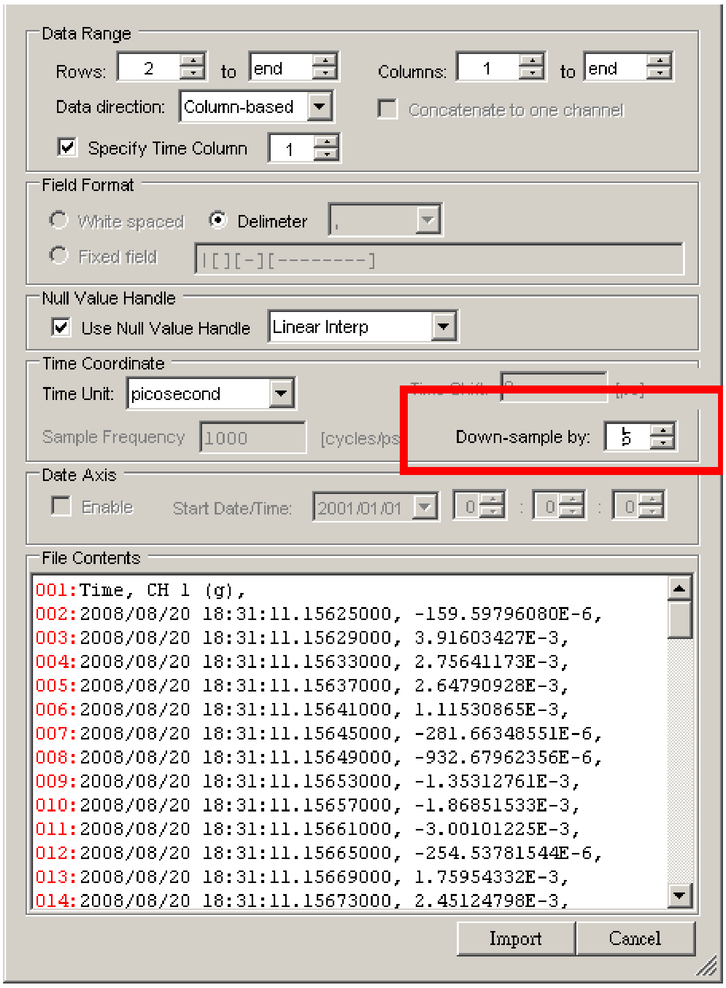


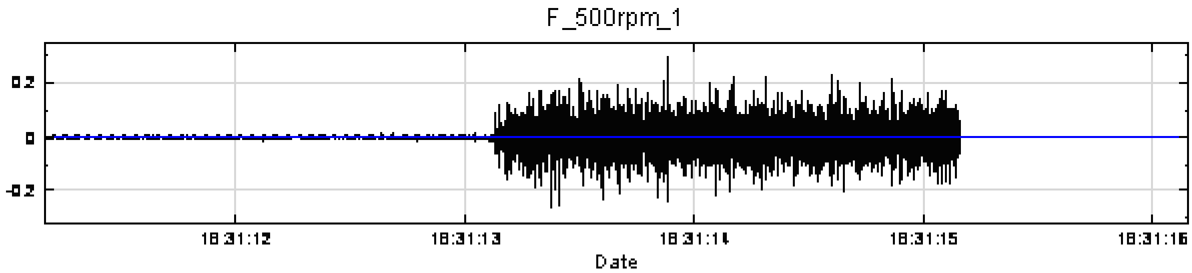
5.1.4 Import csv file format

The data in the csv file format is separated by "," comma character. The first row in the data contains the titles of the columns so the data has to read beginning at second row.



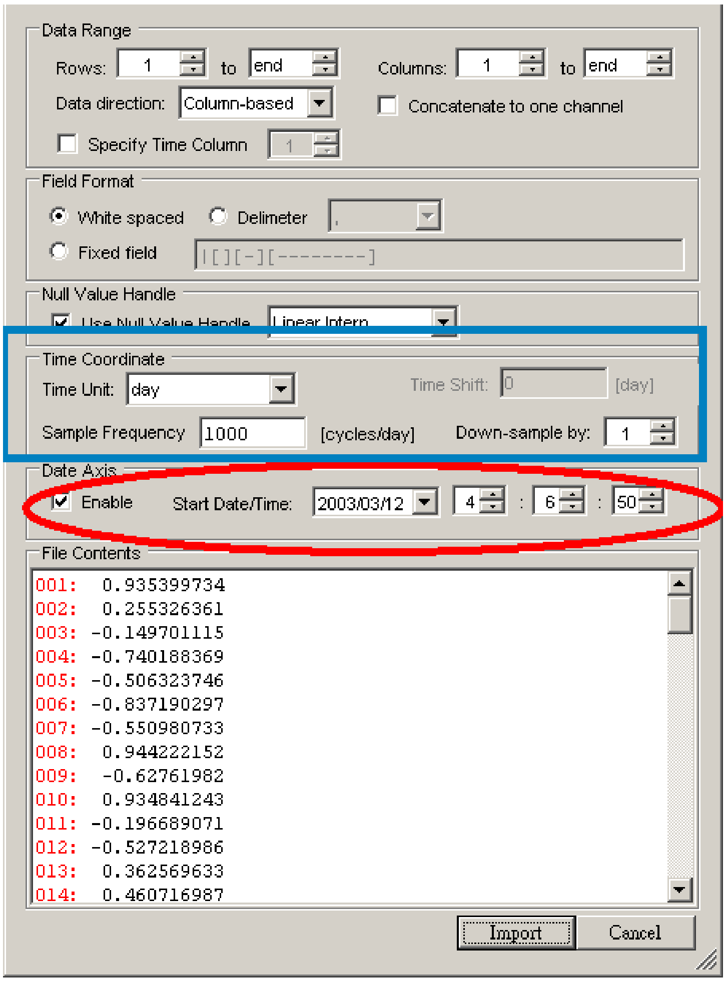
If the data is too large, then you can increase the *Down-Sample* number to 5, so for every 5 data only 1 will be read.

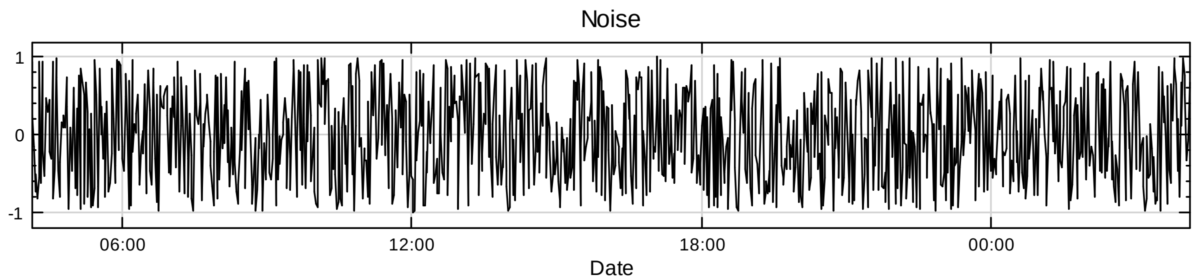




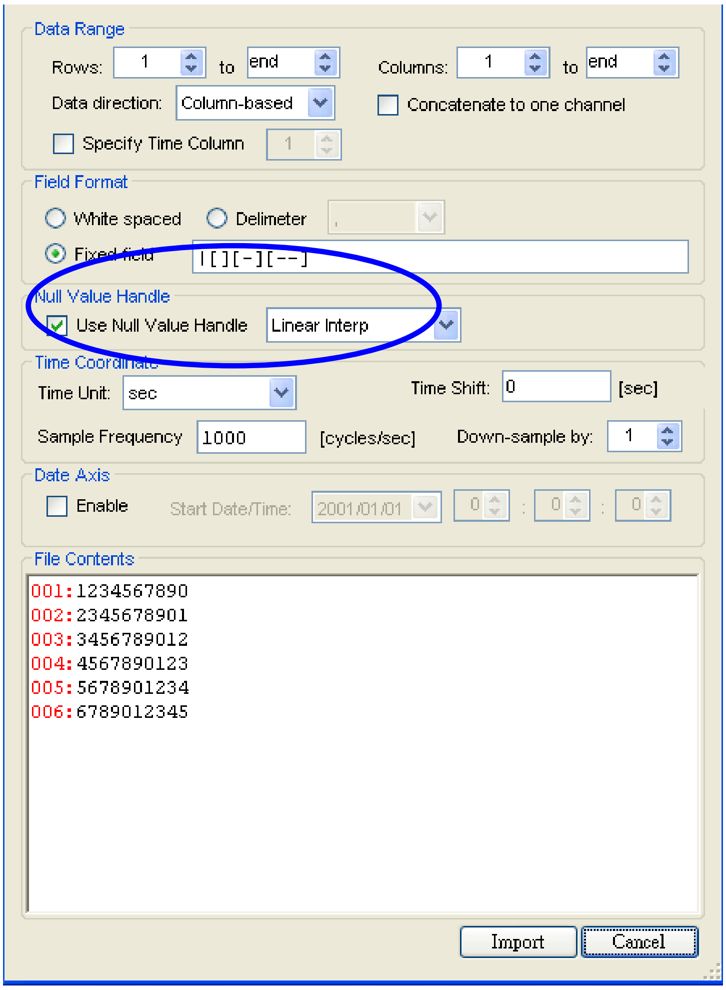
Note: It is not advised to have decimal numbers within the time of the data. E.g. 2005/3/18 15:05:35:.01242

1. If you wish to import a data with date and time and it is not in the csv format then you will have to configure the *Data Axis* and *Time Coordinate* options. In this example, *Time Coordinate* is set as day and *Sample Frequency* is set as 1000. *Data Axis* is Enabled and the date and time is set as 2003/03/12 4(hour):6(minute):50(seconds).





2. If all rows of the imported data have the same character length, you can use customize *Fixed field* to read in the data.

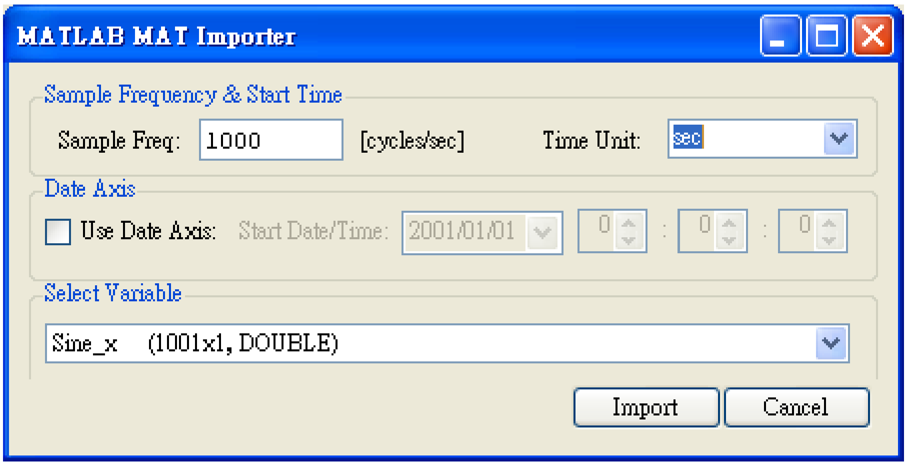


The data imported are placed in four channels: the first channel contains one character, the second channel contains two characters, the third channel contains three characters, and the fourth channel contains four characters. Under *Data Viewer*, the X values are based on the *Sample Frequency* of 1000Hz. So every data value is read at 0.001 increments.

Index	X Value	CH1	CH2	CH3	CH4
0	0	1	23	456	7890
1	0.001	2	34	567	8901
2	0.002	3	45	678	9012

5.1.5 Import Matlab file format

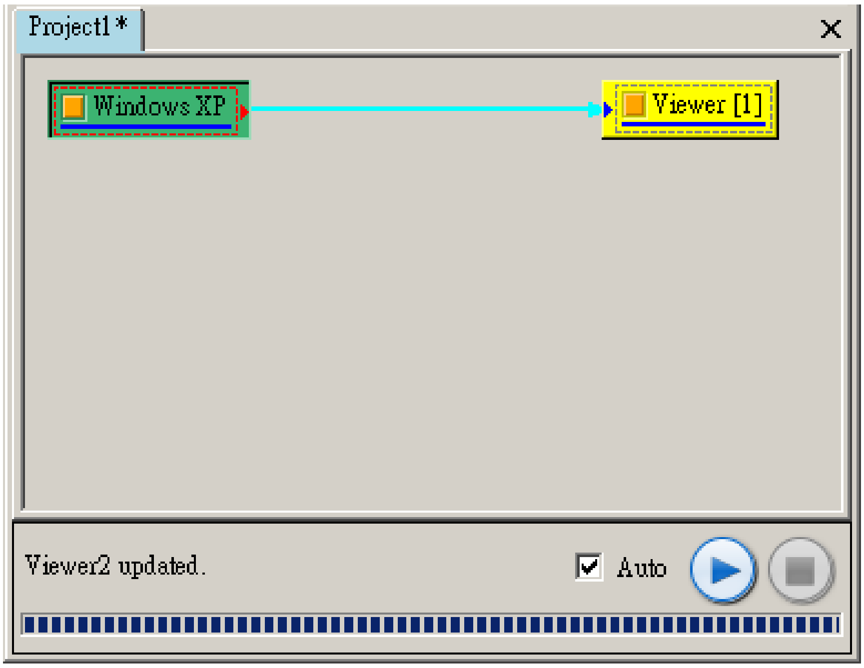
MATLAB MAT Importer will import .mat file format (by MATLAB) which is created in binary format.

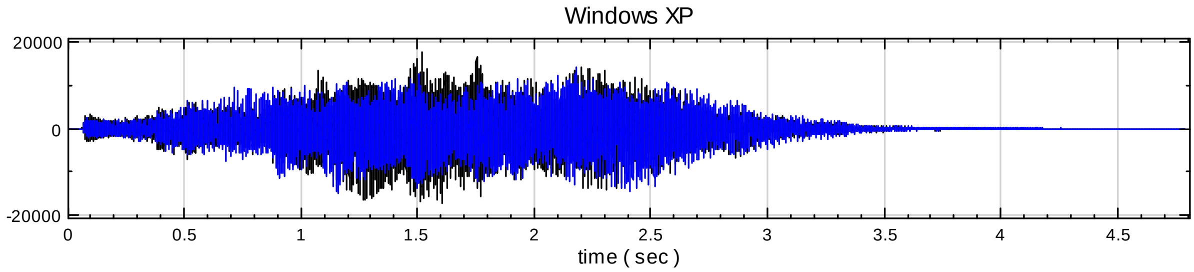


The configuration for the MATLAB MAT Importer is exactly the same as the Text Importer.

5.1.6 Import wav or mp3 file format

If the file imported is .wav or .mp3, these two types of sound format will directly create a Source SFO onto the *Network Workspace* and will not open any file importers.





Property Name	Property Definition	Default Value
Data Range: Contains the options to set the range for the data		
Rows	Enter the range of rows to be read.	1 to End
Columns	Enter the range of columns to be read.	1 to End
Data Direction	Determine the way to read the data, either row based or column based.	Column-based
Concatenate to one channel	Determine if the data is to be displayed in one channel or multiple channels (uncheck)	Unchecked
Specify Time Column	Determine if the time information already exist in the signal data. Check to select the column representing the time information. **NOTE:** After checking the box the data will be displayed in the Indexed format.	Unchecked
Field Format: Contains the options to set how data values are read.		
White spaced	Separate the data values by the white space character.	Yes
Delimiter	Separate the data values by the comma or the TAB character.	No
Fixed Field	Customize your own rules to read the data values.	No
NULL Value Handle: Contains the options to deal with *NULL* or *NaN* values (missing values).		

*NULL*FilledMethod	Select the calculation method to fill in the missing values (please look up Chapter 3.1.4 Fill NULL Value for further details).	Linear Interp
Time Coordinate: Contains the option to set the date and time.		
Time Unit	Select the time unit from psec, nsec, msec, sec, minute, hour, day, week, month (30days) and year (365 days).	sec
Time Shift	Set the starting time of the data.	0
Sample Frequency	Set the Sample Frequency.	1000
Down-sample by	Set the Down-Sample rate. With every increment of the value, the sample data will be shortened to save time during calculation. (**Note:** The Sampling Frequency value will be automatically recalculated depending on the down-sample value. E.g. Sampling Frequency = 1000 with Down-sample = 2 will result in creating an imported Source SFO wtih Sampling Frequency = 500).	1
Date Axis		
Enable	Select to enable the date and time option.	Unchecked→Disabled
Start Date→Time	Set the Date and Time for the data values.	2001/01/01 0:0:0

Related Functions

Viewer, Fill NULL Value, Viewer, Fill NULL Value.

5.2 Noise

Noise is able to create seven different types of Noise signal waves.

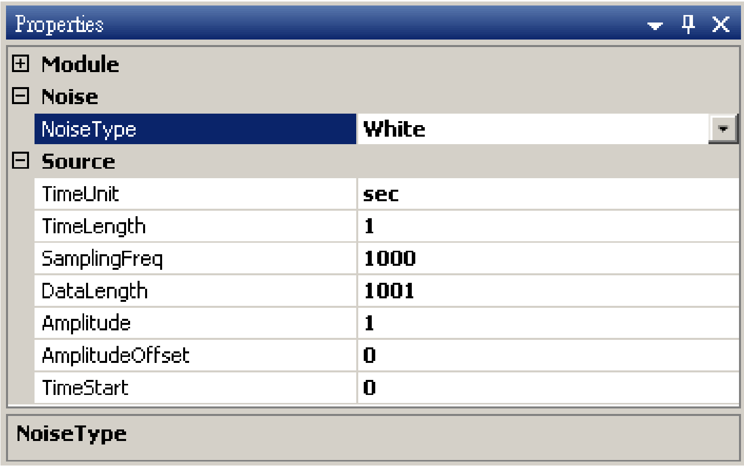
Introduction

Following are the descriptions for each noise definition：

Noise	Equation	Description
White noise	$E[x_{white}] = 0$ $E[f_{white}(t) f_{white}(t-\tau)] = \delta(\tau)$	The noise that has a wide range of frequencies of uniform intensity, where E is expected value. It has an autocorrelation which can be represented by a delta function over the relevant space dimensions.
Gaussian Noise	$x_x = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\bar{t})^2}{2\sigma^2}}$	Gaussian noise is noise that has a probability density function (abbreviated pdf) of the normal distribution (also known as Gaussian distribution).
Speckle Noise	-	Speckle-type noise, its amplitude is either zero or one, it is controlled by the probability P.
Pink Noise	$F[x_{pink}(t)]^2 \propto \frac{1}{f}$	Pink noise or 1/f noise is a signal or process with a frequency spectrum such that the power spectral density is proportional to the reciprocal of the frequency
Brownian Noise	$F[x_{brown}(t)]^2 \propto \frac{1}{f^2}$	Brownian noise is the kind of signal noise produced by Brownian motion hence its alternative name of random walk noise
Blue Noise	$F[x_{blue}(t)]^2 \propto f$	Blue noise's power density increases 3 dB per octave with increasing frequency (density proportional to f) over a finite

		frequency range.
Violet Noise	$F[x_{violet}(t)]^2 \propto f^2$	Violet noise's power density increases 6 dB per octave with increasing frequency (density proportional to f^2) over a finite frequency range

Properties



Property Name	Property Definition	Default Value
TimeUnit	Set the time in ps, ns, us, ms, sec, minute, hour, day, month or year.	sec
TimeLength	Set the value of time selected in TimeUnit.	1
SamplingFreq	Set the number of Sampling frequency (the amount of data values to be sampled).	1000
DataLength	Set the length of the data (SamplingFreq × TimeUnit + 1)	1001
Amplitude	Set the maximum displacement of a periodic wave.	1
AmplitudeOffSet	Set the amplitude offset.	0
TimeStart	Set the start time for the data.	0

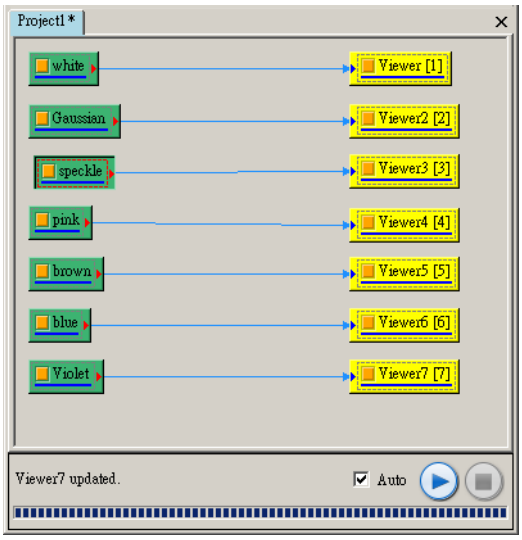
There are two more variable options In Gaussian Noise and Speckle Noise.

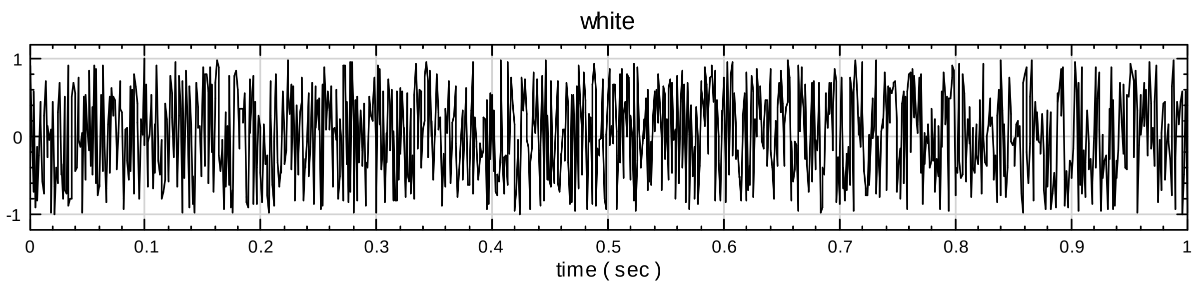
Property Name	Property Definition	Default Value
Sigma(Gaussian)	Set the sigma value for Gaussian Noise.	1
Probability(Speckle)	Set the probability of occurrence for Speckle Noise.	0.005

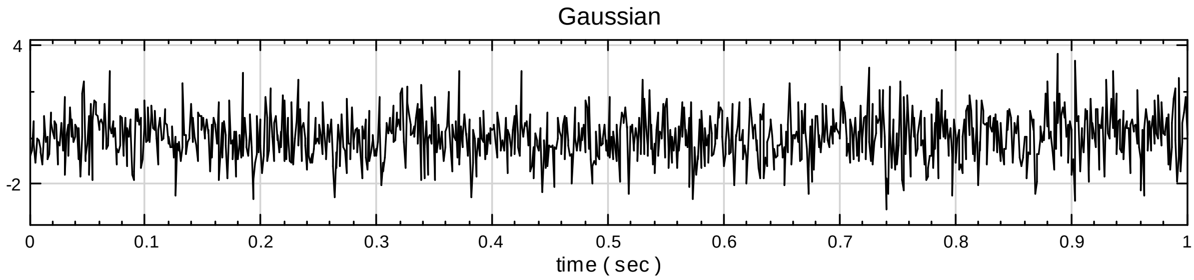
Example

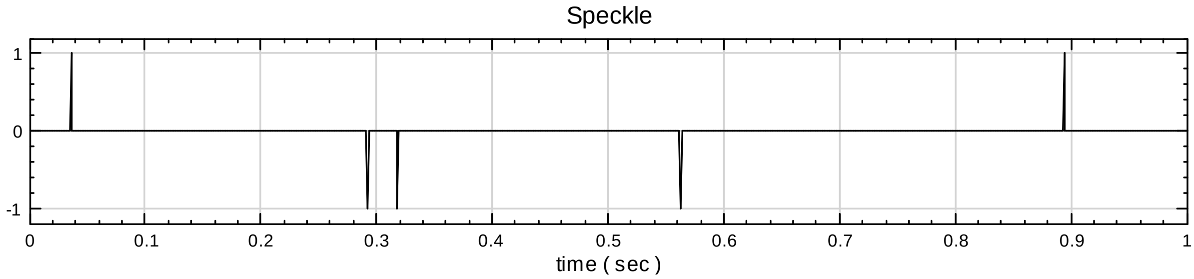
Analysing noise waves:

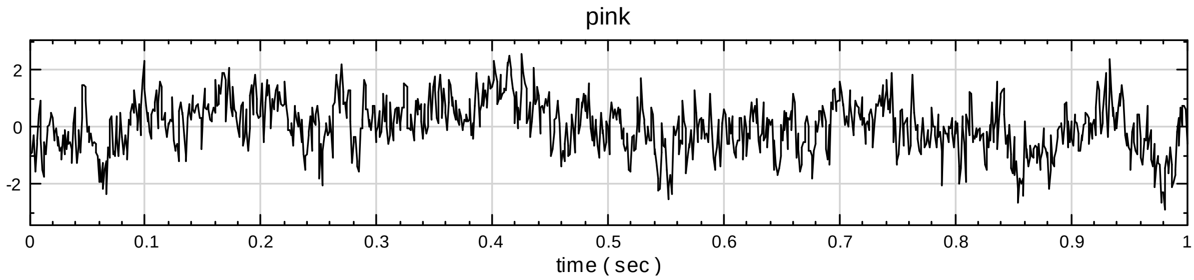
1. Create seven different types of noise through *Source→Noise* and connect each source SFO to a *Viewer→Channel*.

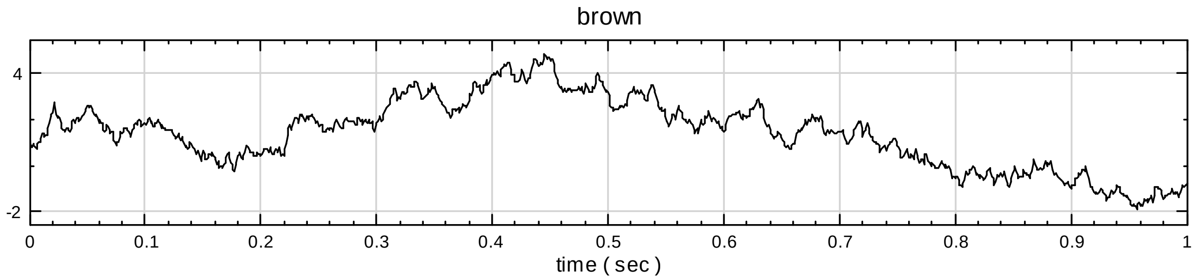


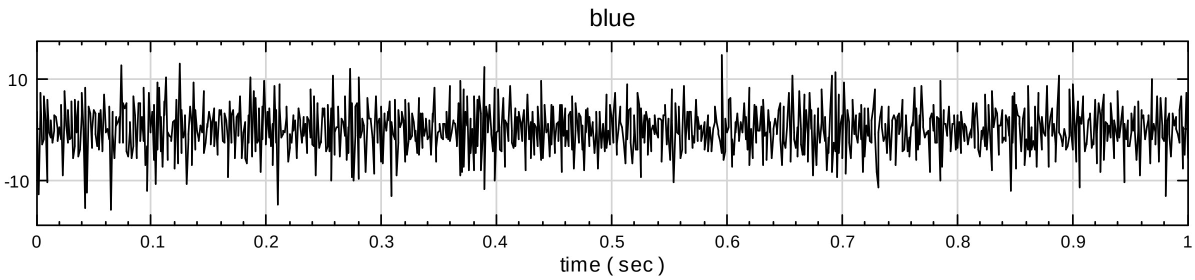


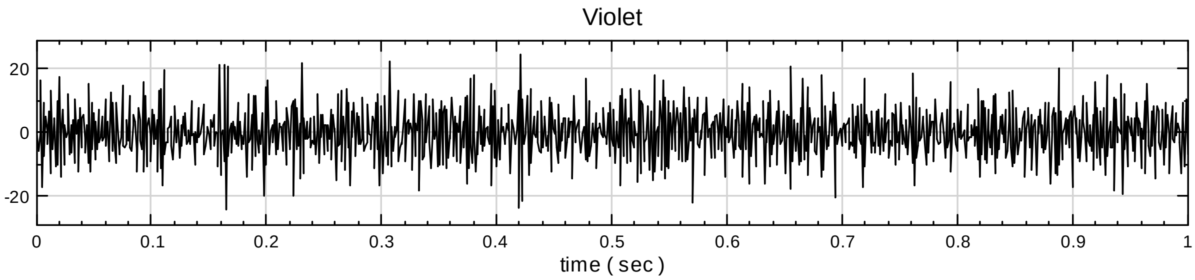

Gaussian
4
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time (sec)


Speckle
1
0
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time (sec)

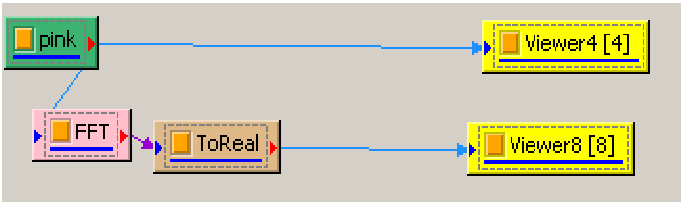

pink
2
0
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time (sec)

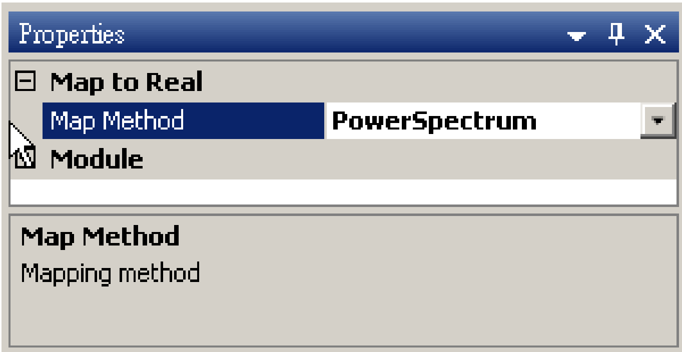

brown
4
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time (sec)

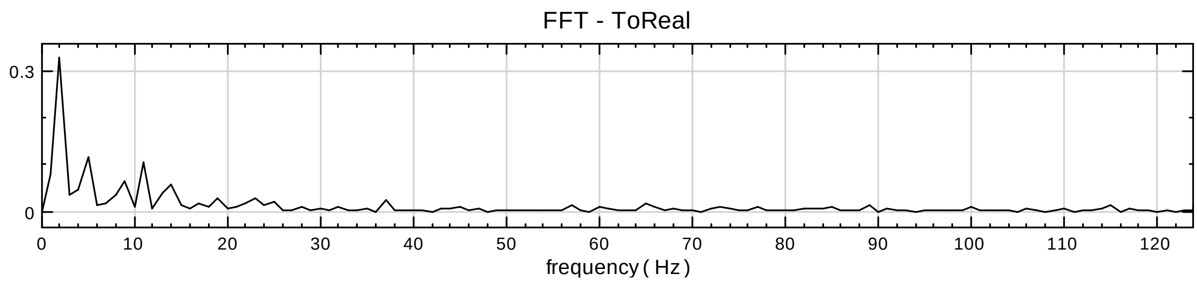

blue
10
-10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time (sec)


Violet
20
0
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time (sec)

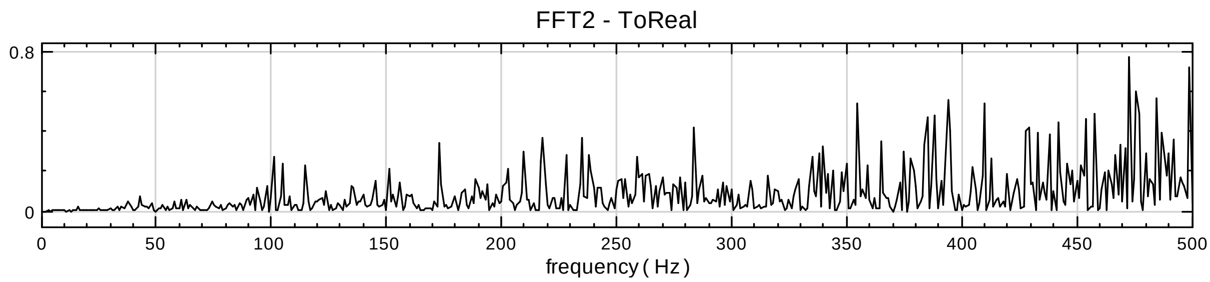
2. Connect the Pink Noise SFO to *Compute→Transform→FFT* and connect the FFT SFO to *Conversion→Map To Real* and finally connect the result to a viewer SFO. Change the *Properties/MapMethod* of the ToReal SFO to *PowerSpectrum*. You can observe that the power spectrum density increases as the frequency decreases.







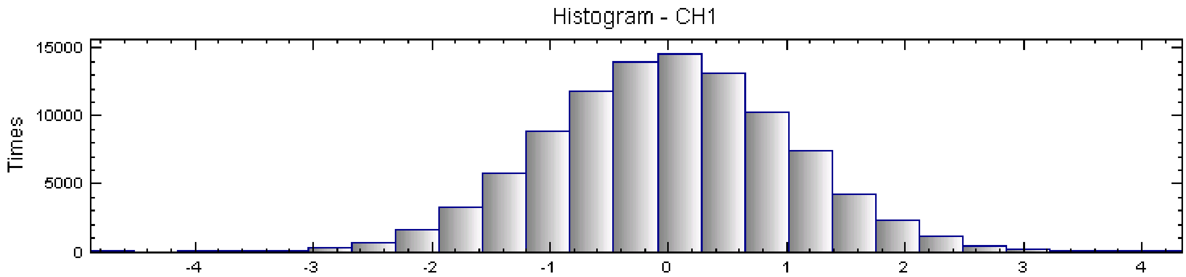
3. Repeat the steps to the other source signals. The Blue noise graph is shown below after the steps.



You can observe that the power spectrum density increases as frequency increases.

4. Set the Noise Type to Gaussian Noise and set the *Time Length* to 100

seconds and view it with *Viewer→Histogram Viewer*.



Related Functions

Channel Viewer, Fourier Transform, Map To Real.

Reference

http://en.wikipedia.org/wiki/Colors_of_noise

http://en.wikipedia.org/wiki/Gaussian_noise

5.3 Sine Wave

Explanation is given here for the Source→Sine Wave SFO.

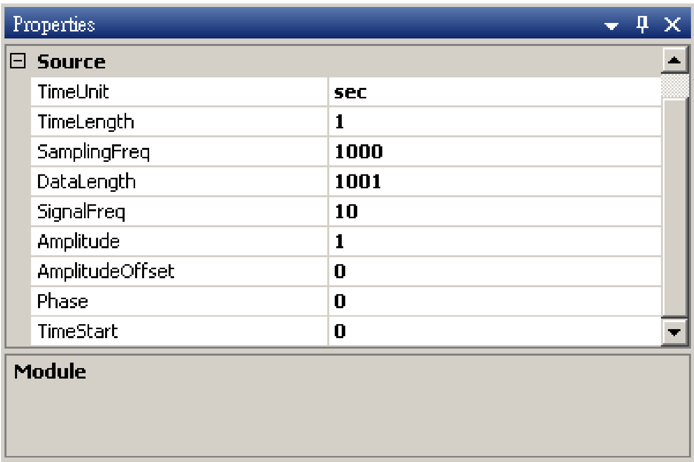
Introduction

Let *t*=time, *N*=length of the signal, $t_i = [t_0, t_1 \ldots t_{N-1}]$ is the representation of the time coordinate and sine wave can be generated by

$$x_i = A \cdot \sin(\omega t_i + \delta) + V_0$$

Where A=amplitude, ω=angular frequency, δ=phase at t_0, V_0= offset from X axis, and sampling frequency is defined as $f = \dfrac{\omega}{2\pi}$.

Properties



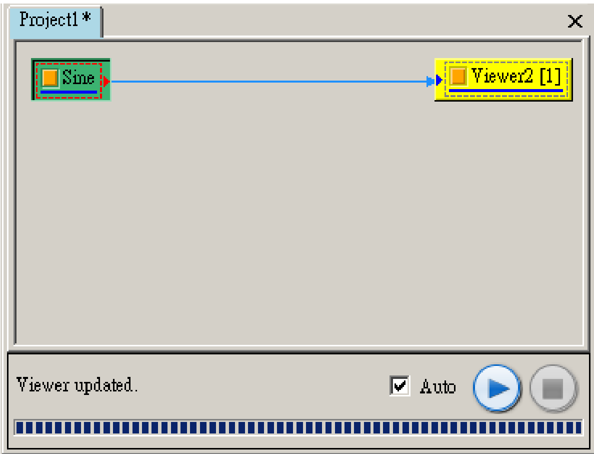
Property Name	Property Definition	Default Value
TimeUnit	Set the time in ps, ns, us, ms, sec, minute, hour, day, month or year.	sec
TimeLength	Set the value of time selected in TimeUnit.	1
SamplingFreq	Set the number of Sampling frequency (the amount of data values to be sampled).	1000
DataLength	Set the length of the data (SamplingFreq × TimeUnit + 1)	1001

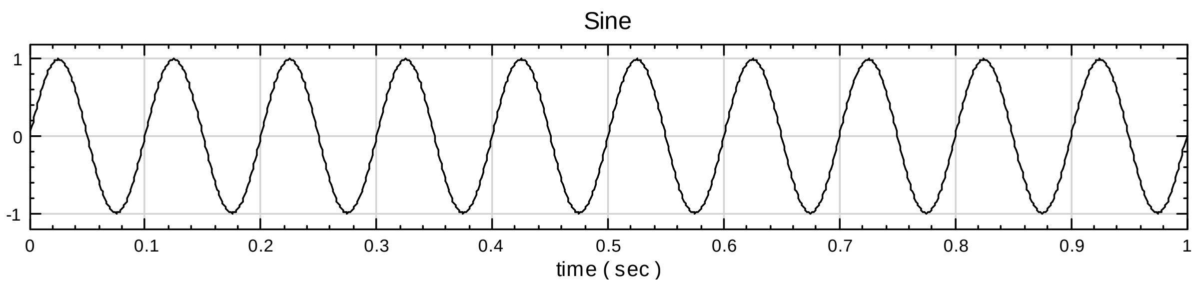
Amplitude	Set the maximum displacement of a periodic wave.	1
AmplitudeOffSet	Set the amplitude offset.	0
TimeStart	Set the start time for the data.	0
Phase	Set the Phase in degree. When the phase is non-zero, the entire waveform appears to be shifted in time with specified value.	0°

Example

Create a Sine wave.

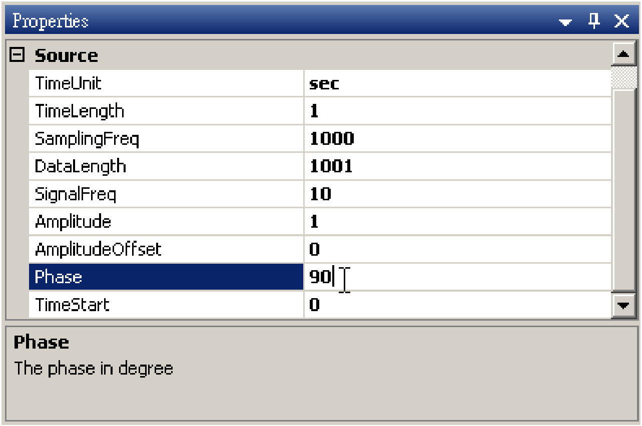
1. Create *Source*→*Sine Wave*.

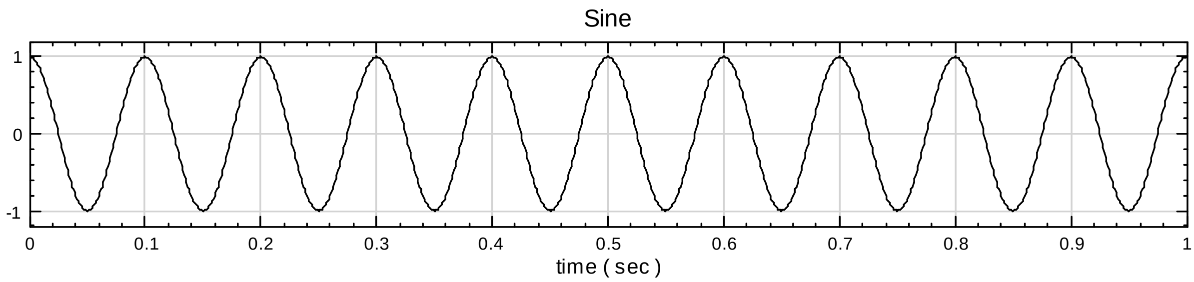




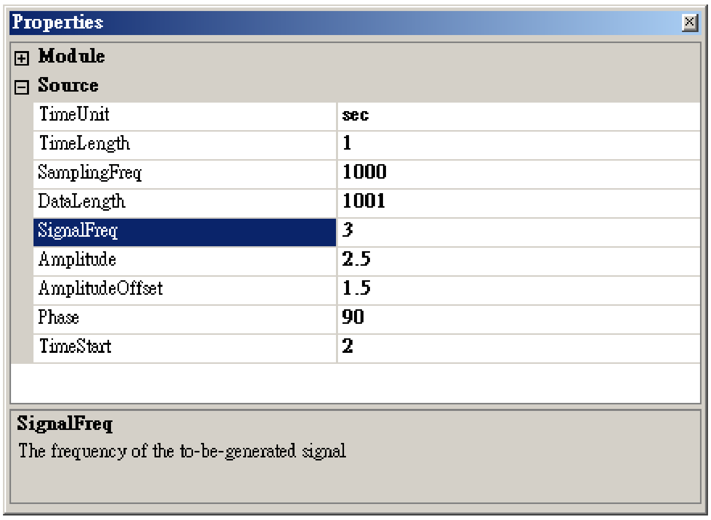
In *Properties/Source/SamplingFreq* is set to 1000 and *SignalFreq* is set to 10.

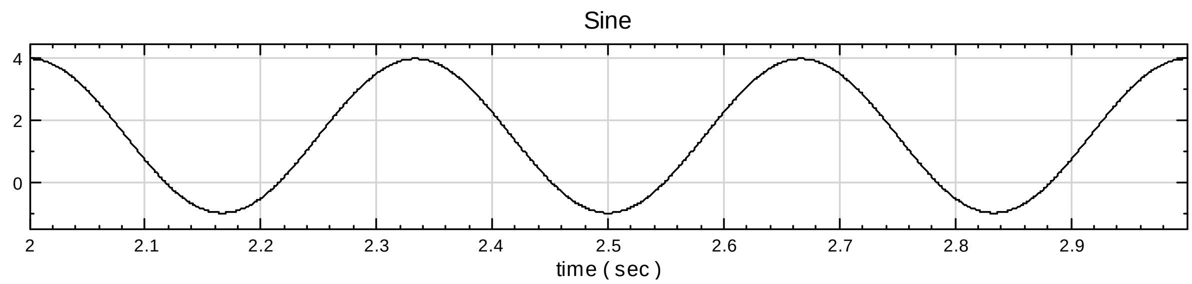
2. If you set the *Properties/Source/Phase* to 90° then the Sine Wave will become a Cosine Wave.





3. Set the *Properties/Source/SignalFreq* to 3, *Amplitude* to 2.5, *AmplitudeOffSet* to 1.5, and *TimeStart* to 2, the graph is shown in the image below.





Related Functions

Viewer, Square, Triangle, Viewer, Square, Triangle.

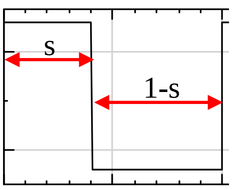
5.4 Square Wave

Explanation is given for the *Source→Square Wave* SFO.

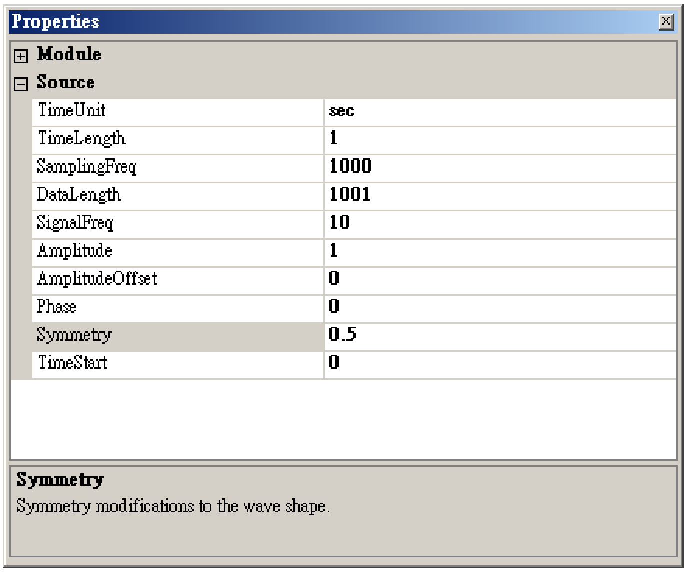
Introduction

$$x_{sqr}(t)=\begin{cases} A+V_0, & \delta \le t < (\frac{1}{f})s+\delta \\ -A+V_0, & (\frac{1}{f})s+\delta \le t < (\frac{1}{f})+\delta \end{cases}, x_{sqr}(t+T)=x_{sqr}(t)$$

Where A=amplitude, f = sampling frequency, δ = phase at t_0, V_0 = offset from X axis, the ratio s is shown in the image below.



Properties



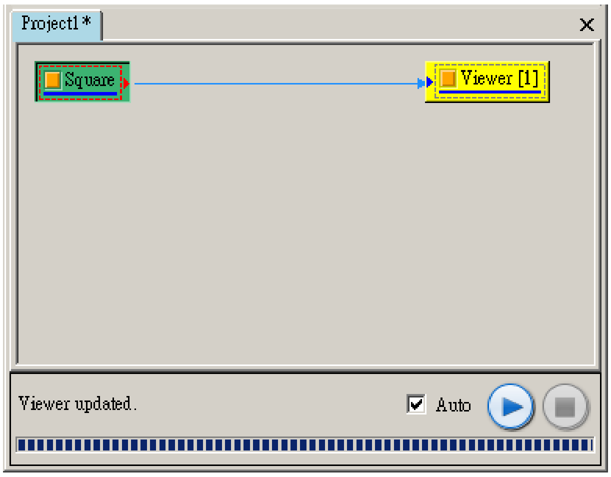
Property Name	Property Definition	Default Value
TimeUnit	Set the time in ps, ns, us, ms, sec, minute,	Sec

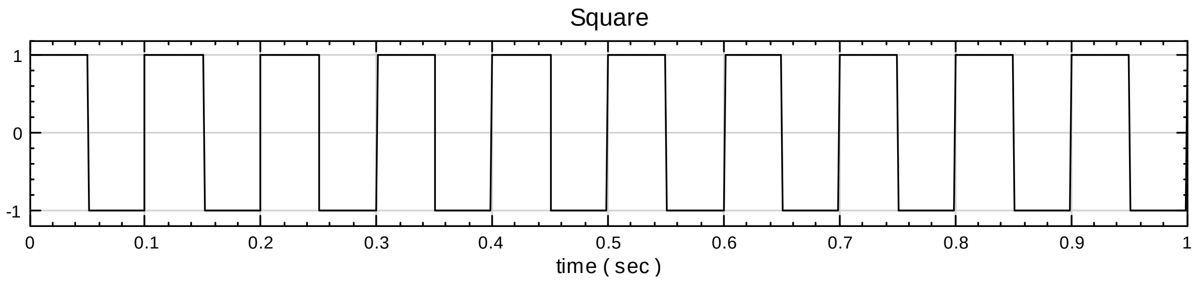
	hour, day, month or year.	
TimeLength	Set the value of time selected in TimeUnit.	1
SamplingFreq	Set the number of Sampling frequency (the amount of data values to be sampled).	1000
DataLength	Set the length of the data (SamplingFreq × TimeUnit + 1)	10001
Amplitude	Set the maximum displacement of a periodic wave.	10
AmplitudeOffSet	Set the amplitude offset.	0
TimeUnit	Set the time in ps, ns, us, ms, sec, minute, hour, day, month or year.	1
TimeLength	Set the value of time selected in TimeUnit.	0
Phase	Set the Phase in degree. When the phase value is non-zero, the entire waveform appears to be shifted in time by the entered amount.	0°
Symmetry	Symmetry set at 0.5 is equal symmetry where the left of the inflection point takes up 0.5 (half) of the period. E.g. Symmetry=0.2 means that the left of the inflection point takes up only one-fifth of the period.	0.5
TimeStart	Set the start time for the data.	0

Example

Create a Square wave.

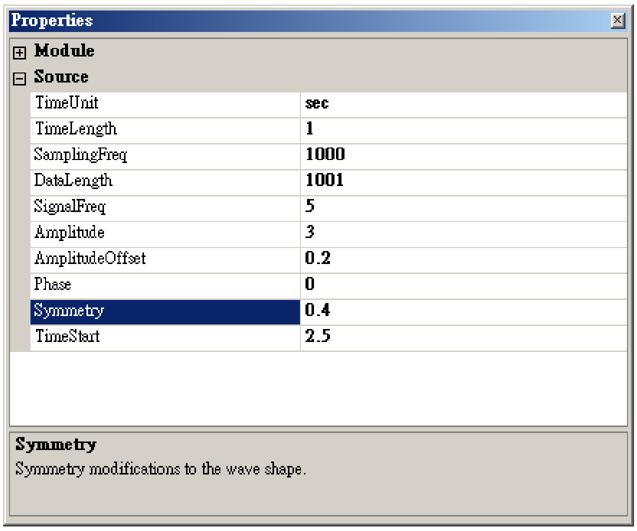
1. Create *Source*→*Square Wave*.

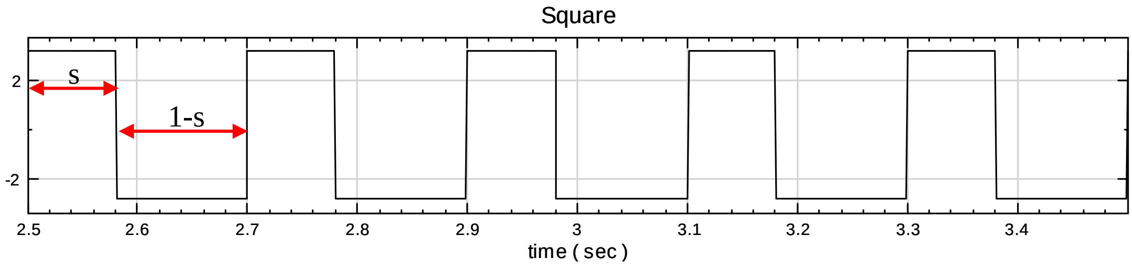




In *Properties/Source/SamplingFreq* is set to 1000 and *SignalFreq* is set to 10.

2. Set the *Properties/Source/SignalFreq* to 5, *Amplitude* to 3, *AmplitudeOffSet* to 0.2, *Phase* to 0, *Symmetry* to 0.4, *TimeStart* to 2.5, the graph is shown in the image below.





Related Functions

Viewer

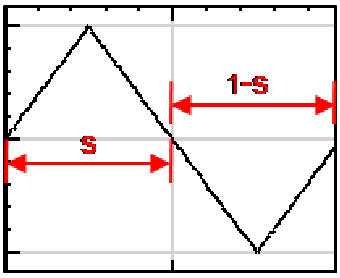
5.5 Triangle Wave

Explanation is given for the *Source→Triangle Wave*.

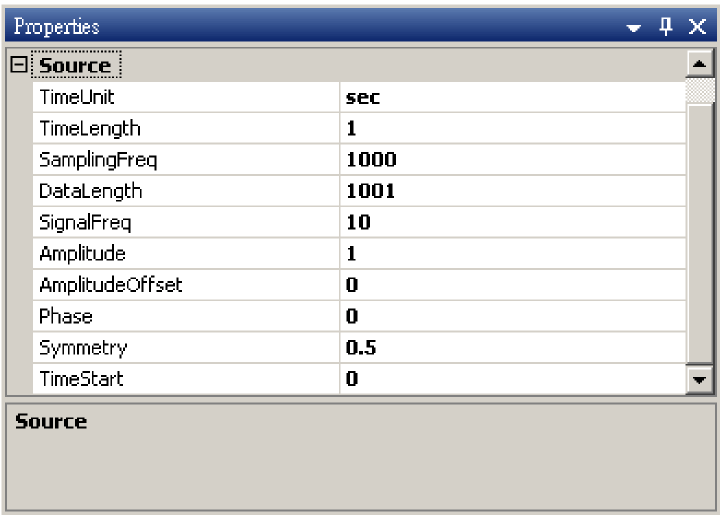
Introduction

$$x_{sqr}(t)=\begin{cases} \dfrac{Af}{s}t+V_0, & \theta \le t < (\dfrac{1}{f})s+\theta \\ A\left[1-\dfrac{f}{1-s}(t-\dfrac{s}{f})\right]+V_0, & (\dfrac{1}{f})s+\theta \le t < (\dfrac{1}{f})+\theta \end{cases} \quad ,x_{sqr}(t+T)=x_{sqr}(t)$$

Where A=amplitude, f = sampling frequency, δ = phase at t_0, V_0 = offset from X axis, the ratio s is shown in the image below.



Properties



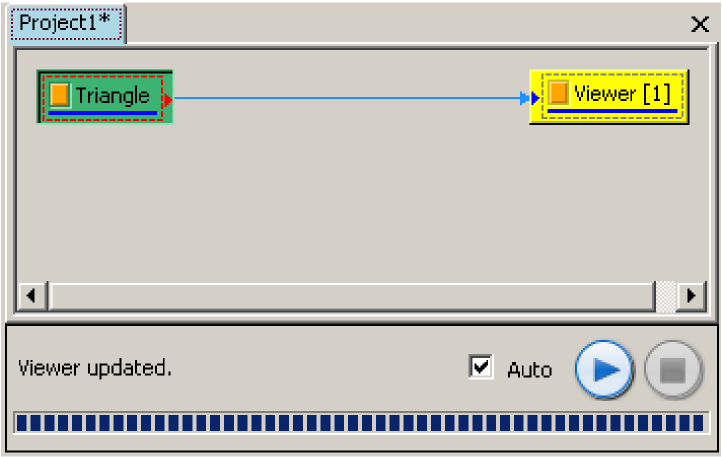
Source	
TimeUnit	sec
TimeLength	1
SamplingFreq	1000
DataLength	1001
SignalFreq	10
Amplitude	1
AmplitudeOffset	0
Phase	0
Symmetry	0.5
TimeStart	0

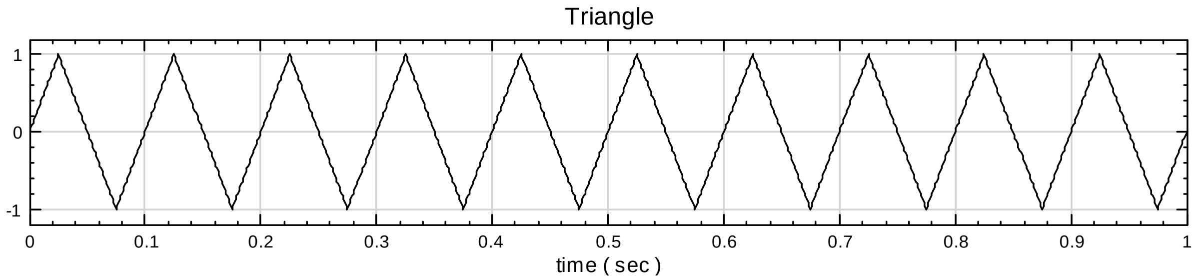
Property Name	Property Definition	Default Value
TimeUnit	Set the time in ps, ns, us, ms, sec, minute, hour, day, month or year.	Sec
TimeLength	Set the value of time selected in TimeUnit.	1
SamplingFreq	Set the number of Sampling frequency (the amount of data values to be sampled).	1000
DataLength	Set the length of the data (SamplingFreq × TimeUnit + 1)	10001
Amplitude	Set the maximum displacement of a periodic wave.	10
AmplitudeOffSet	Set the amplitude offset.	0
TimeUnit	Set the time in ps, ns, us, ms, sec, minute, hour, day, month or year.	1
TimeLength	TimeUnit. Set the value of time selected in TimeUnit.	0
Phase	Set the Phase in degree. When the phase is non-zero, the entire waveform appears to be shifted in time by the entered amount.	0°
Symmetry	Symmetry set at 0.5 is equal symmetry where the left of the inflection point takes up 0.5 (half) of the period. E.g. Symmetry=0.2 means that the left of the inflection point takes up only one-fifth of the period.	0.5
TimeStart	Set the start time for the data.	0

Example

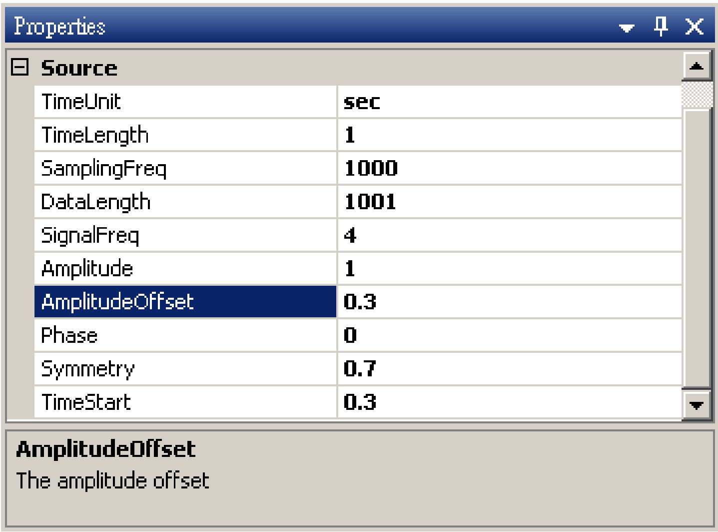
Create a Triangle wave.

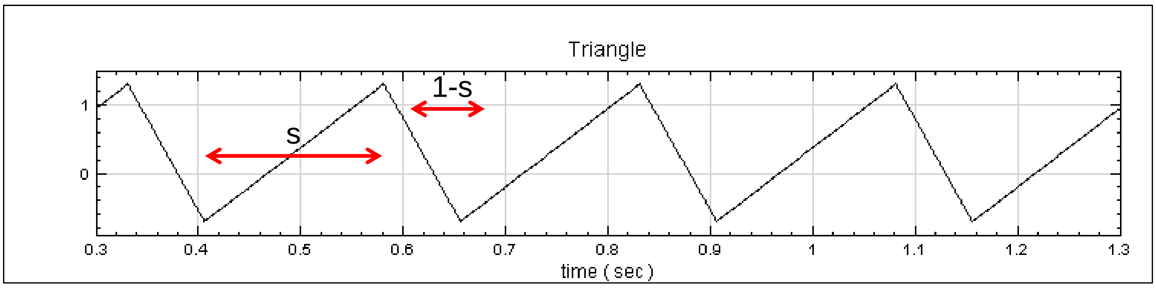
1. Create *Source→Triangle Wave*.





2. Set the *Properties/Source/SignalFreq* to 4, *Amplitude* to 1, *AmplitudeOffSet* to 0.3, *Phase* to 0, *Symmetry* to 0.7 and *TimeStart* to 0.3, the graph is shown in the image below.





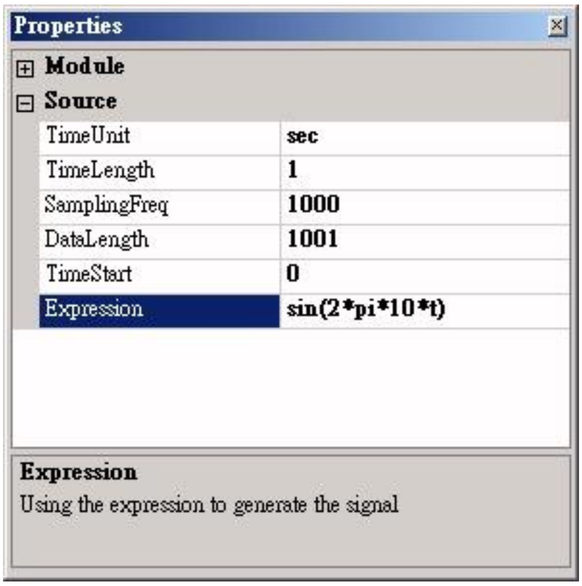
Related Functions

Square Wave, Sine Wave, Viewer.

5.6 Custom Wave

The users can input equations to create signals via this module.

Properties



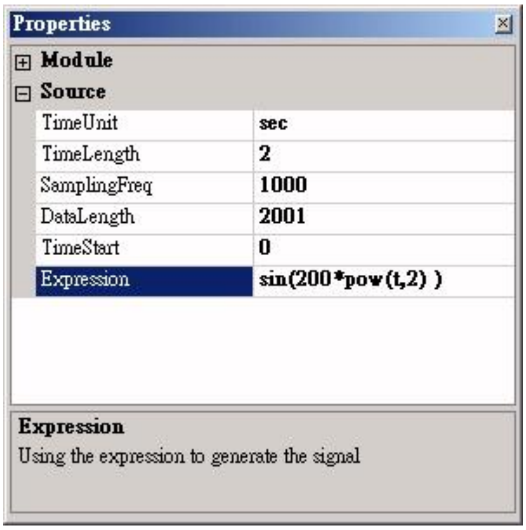
Property Name	Property Definition	Default Value
TimeUnit	The unit of time: second, minute, hour, day, month, year.	second
TimeLength	The length of time.	1
SamplingFreq	The frequency of sampling.	1000
DataLength	The total length of the data, i.e. the number of sampling points. It equals SamplingFrequency * TimeLength + 1.	10001
TimeStart	The start point in time.	0
Expression	Set the equations to calculate the signal.	

where expression can use sin, cos, tan, exp and asin etc math functions, which are the same as functions in the fn menu of Math module. (Please also refer to math functions for C# language). Please note the expression of power, a^b is written as pow(a,b).

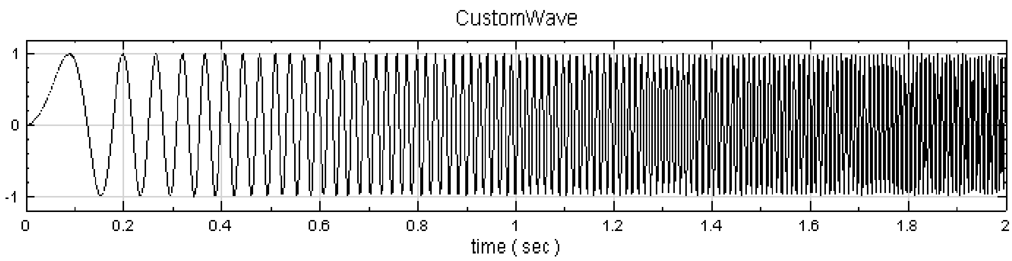
Example

Build a quasi-steady signal in which the time is direct ratio with the frequency:

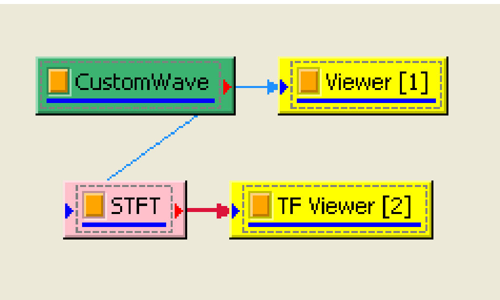
Under the menu of Source / Custom Wave, the user set the TimeLength to be 2. The setting of the Expression is as below:

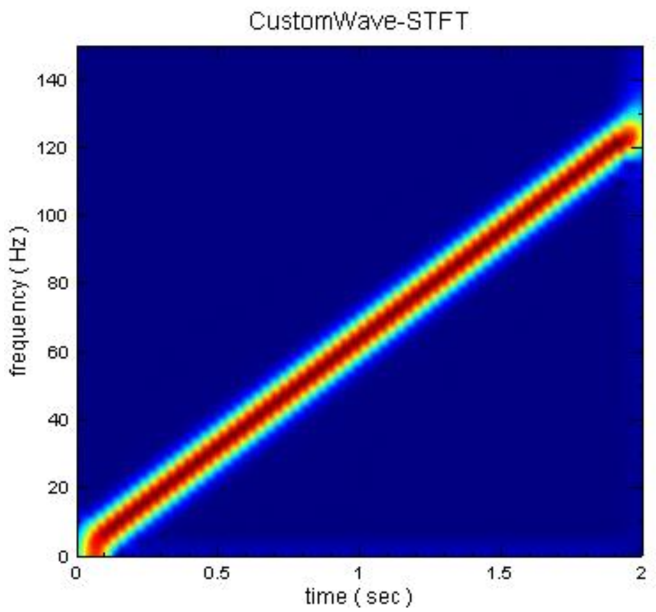


Viewing the function with Viewer as below:



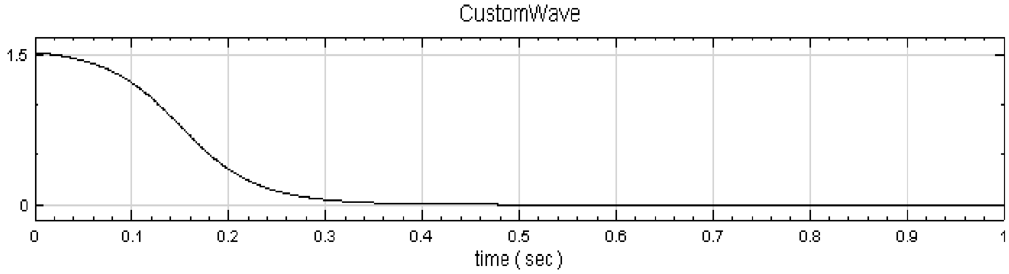
Then, the user can validate that the frequency has direct ratio with time using the short-term Fourier transform. The setting and the frequency-time figure is shown as below:





The user can also build a wave of $\tan^{-1}(e^{3-20\cdot t})$:

TimeUnit	sec
TimeLength	1
SamplingFreq	1000
DataLength	1001
TimeStart	0
Expression	atan(exp(3-20*t))



Related Functions

Channel Viewer, ShortTerm Fourier Transformation, Math.

References

http://msdn.microsoft.com/en-us/library/system.math_methods.aspx

5.7 Advanced

5.7.1 Impulse (Professional Only)

The user can build a pulse signal with the Impulse module.

Introduction

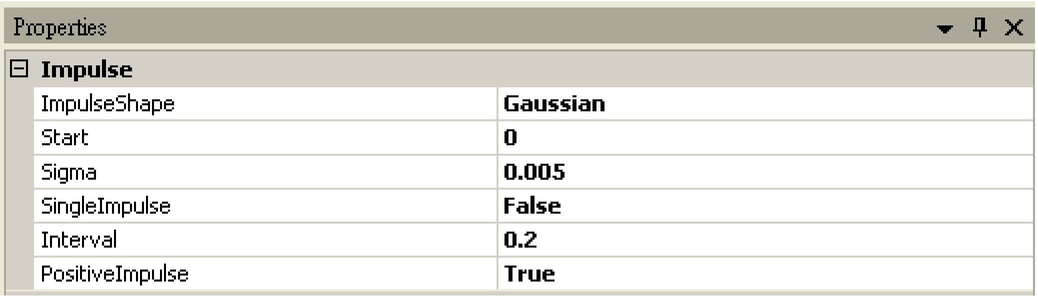
Given t as the time, a series with N elements is $t_i = \{t_0, t_1, \dots t_{N-1}\}$. A perfect pulse signal X can be presented as:

$$X(t) = A \cdot u(t_i - T) + V_0 \; ; \begin{cases} u(0) = 1 \\ u(t) = 0 \quad t \neq 0 \end{cases}$$

where T is the starting time of the pulse, A is the amplitude, V_0 is the offset of the amplitude. In this module, the user can implement three kinds of pulse signals, including Gaussian, Square and Decay signals.

TYPE	DEFINETION	INTRODUCTION
Gaussian	$\delta_a(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-t_0)^2}{2\sigma^2}}$	This signal presents the Normal Distribution in time domain. t_0 is the starting time of the pulse signal and σ is the sharpness of the pulse signal.
Square	$\delta(t) = \begin{cases} A \, , 0 \le t - t_0 < w \\ 0 \, , \, otherwise \end{cases}$	This signal is approximate to a square wave with tiny width. t_0 is the starting time of the pulse signal and W is the width of the square pulse wave.
Decay	$\delta(t) = \begin{cases} A \, , & 0 \le t - t_0 \le w \\ Ae^{t_0 + w - t} \, , & otherwise \end{cases}$	This signal is approximate to a square wave with tiny width and decays in the exponential ratio. t_0 is the starting time of the pulse signal and W is the width of the square pulse wave.

Properties



Property Name	**Property Definition**	**Default Value**
ImpulseShape	The shape of the pulse: Gaussian, Square, Decay.	Gaussian
Start	The starting time of the pulse.	0
SingleImpulse	To create a single pulse.	False
Interval	The interval between two pulses.	0.2
PositiveImpulse	Set pulse to be positive.	True

If ImpulseShape = Gaussian, one more property is:

Property Name	**Property Definition**	**Default Value**
Sigma	Set the width of the Normal Distribution. The smaller of sigma, the sharper of the shape.	0.005

If ImpulseShape = Square, one more property is:

Property Name	**Property Definition**	**Default Value**
Width	Set the width of square waves. If width is 0, the actual width is the time distance between two points of input signal.	0

If ImpulseShape = Decay, one more property is:

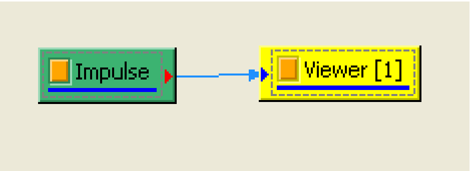
Property	**Property Definition**	**Default**

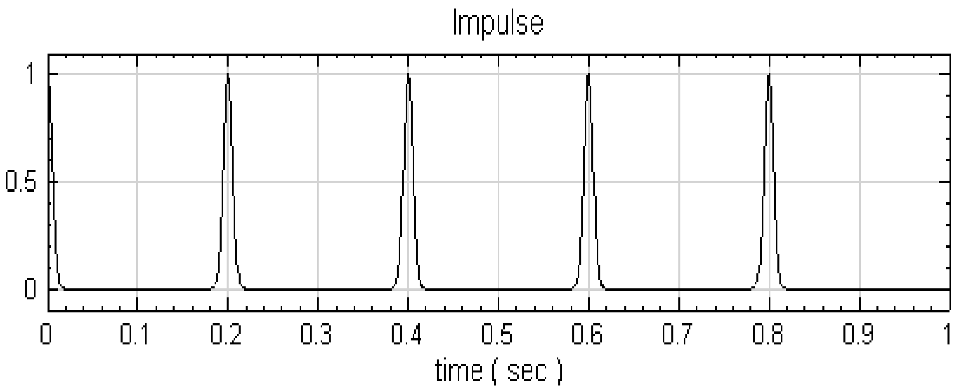
Name		Value
Width	Set the width of square waves. If width is 0, the actual width is the time distance between two points of input signal.	0
Decay	Set the decay time of the amplitude. The smaller the value, the faster the decay.	0.005

Example

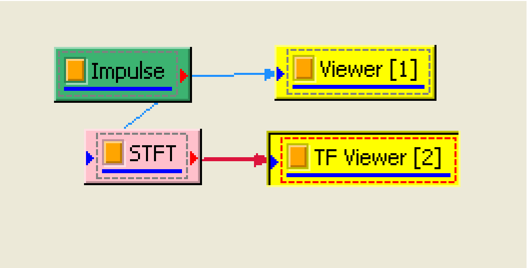
Create and analyze a signal:

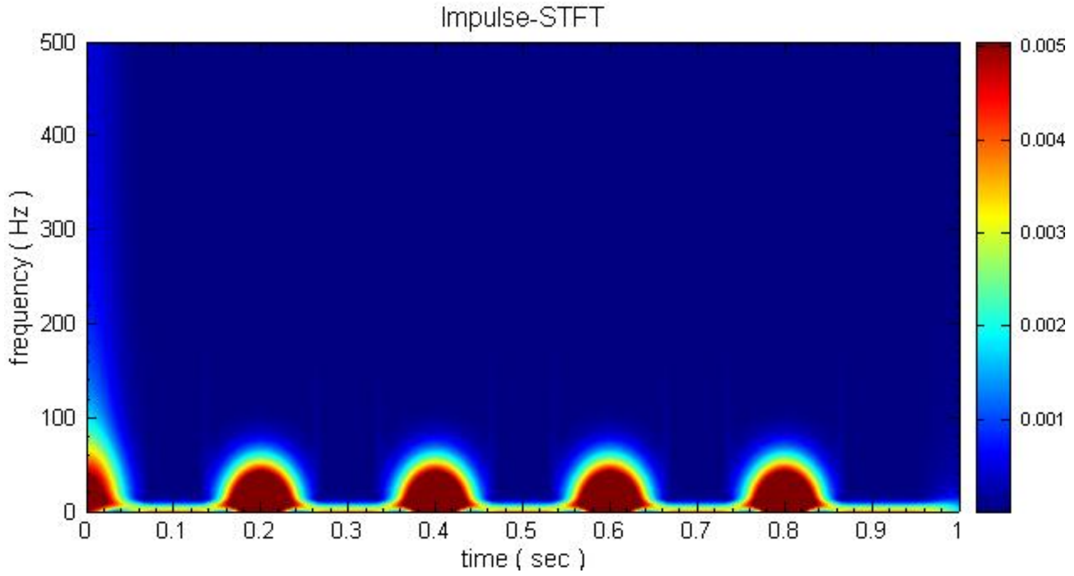
Create a pulse signal with the Source / Advanced / Impulse and viewing its results with the Viewer / Channel Viewer.



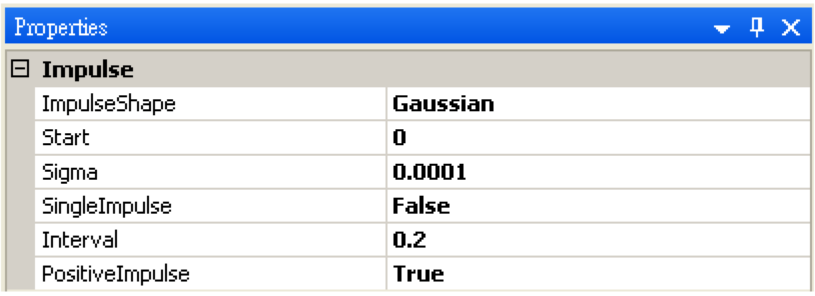


Connect the Impulse to the Compute / TFA / Short Term Fourier Transform and view the result with the Viewer / Time-Frequency Viewer.



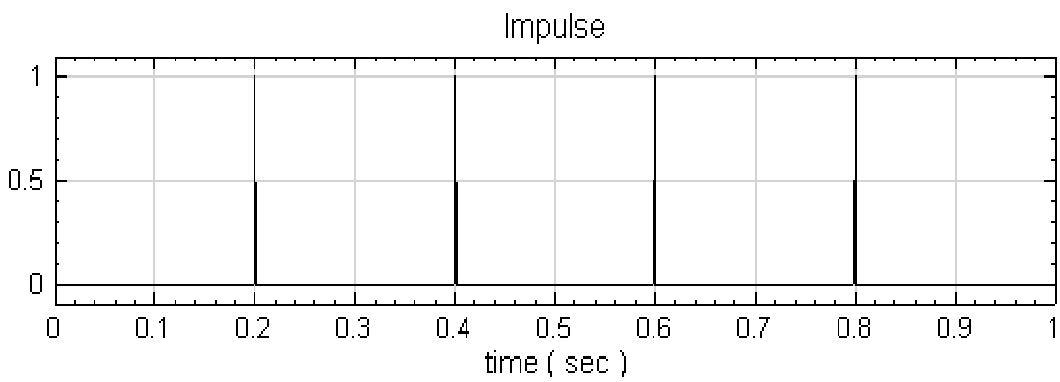


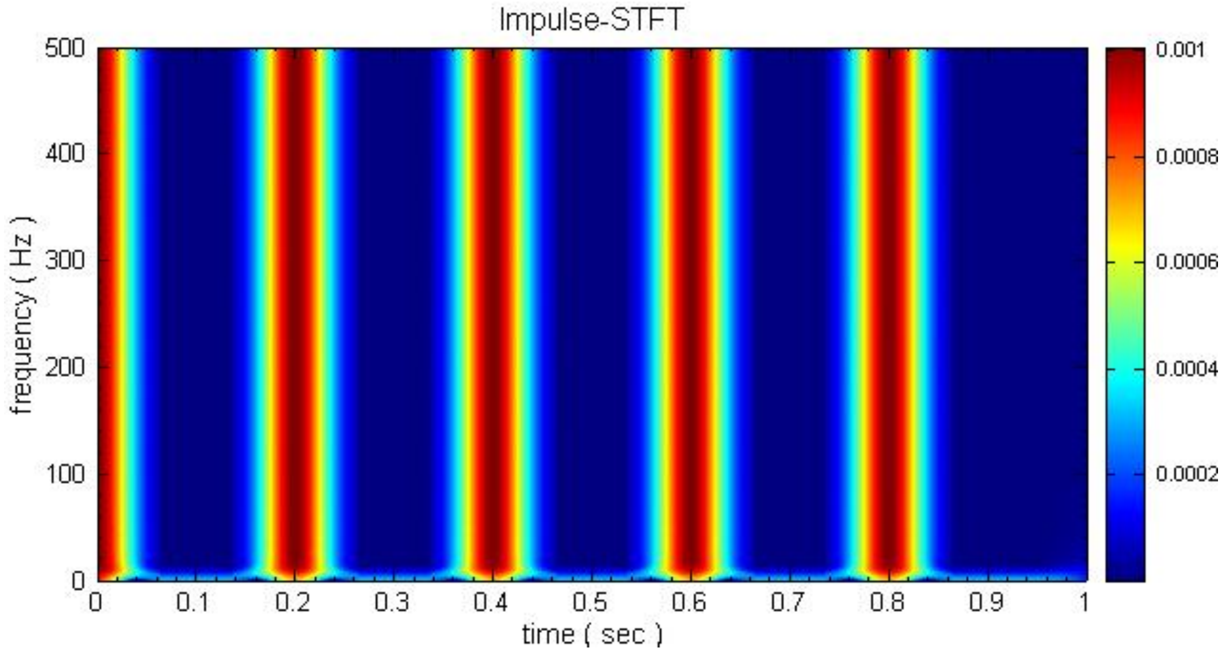
Set the properties of the Impulse. Set Sigma=0.0001. The smaller is the value of Sigma, the better is the pulse. The pulse is almost perfect as shown in Channel Viewer. It clearly shows that frequency band is quite large in TFA Viewer.



Properties

Impulse	
ImpulseShape	**Gaussian**
Start	**0**
Sigma	**0.0001**
SingleImpulse	**False**
Interval	**0.2**
PositiveImpulse	**True**





Related Functions

Channel Viewer, Time-Frequency Viewer, Short Term Fourier Transform.

5.7.2 Jaehne (Professional Only)

Build Jaehne signal.

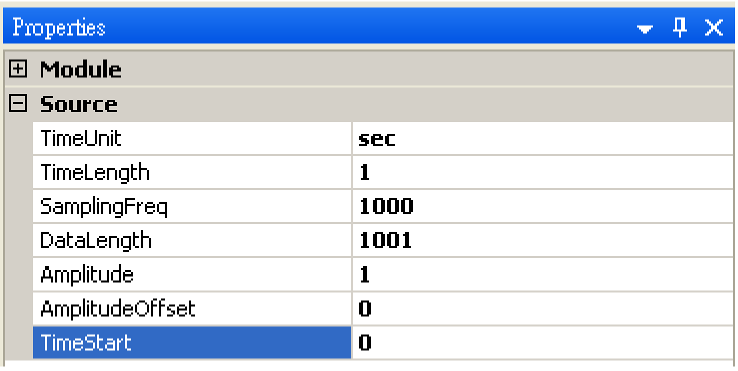
Introductions

Given t as the time, a series with N elements is $t_i = \{t_0, t_1, \dots t_{N-1}\}$. The Jaehne signal is:

$$X(t_i) = A \times sin(\frac{0.5\pi * i^2}{N}) + V_0$$

where A is the amplitude and V_0 is the offset of the amplitude.

Properties



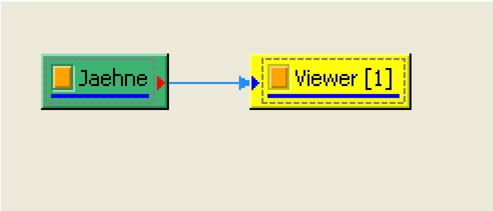
Property Name	**Property Definition**	**Default Value**
TimeUnit	The unit of time: ps, ns, us, ms, sec, minute, hour, day, month, year.	sec
TimeLength	The length of time.	1
SamplingFreq	The frequency of sampling, i.e. data points in unit time.	1000
DataLength	The total length of the data, i.e. the number of sampling points. It equals SamplingFrequency * TimeLength + 1.	1001
Amplitude	The amplitude of the signal.	1
AmplitudeOffSet	The offset of the amplitude.	0

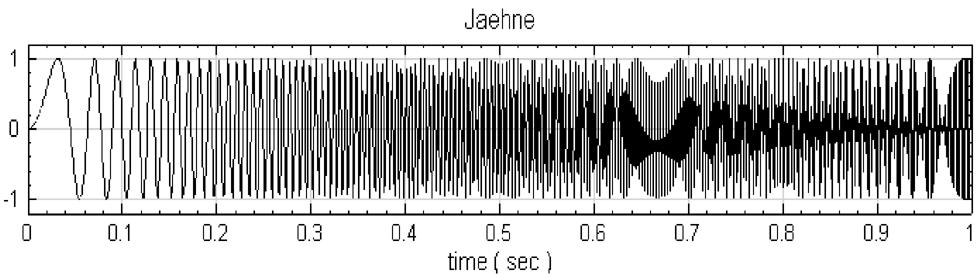
TimeStart	The starting time of the signal.	0

Example

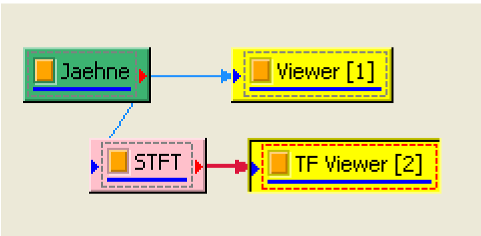
Build the Jaehne signal and do a time-frequency analysis of the signal:

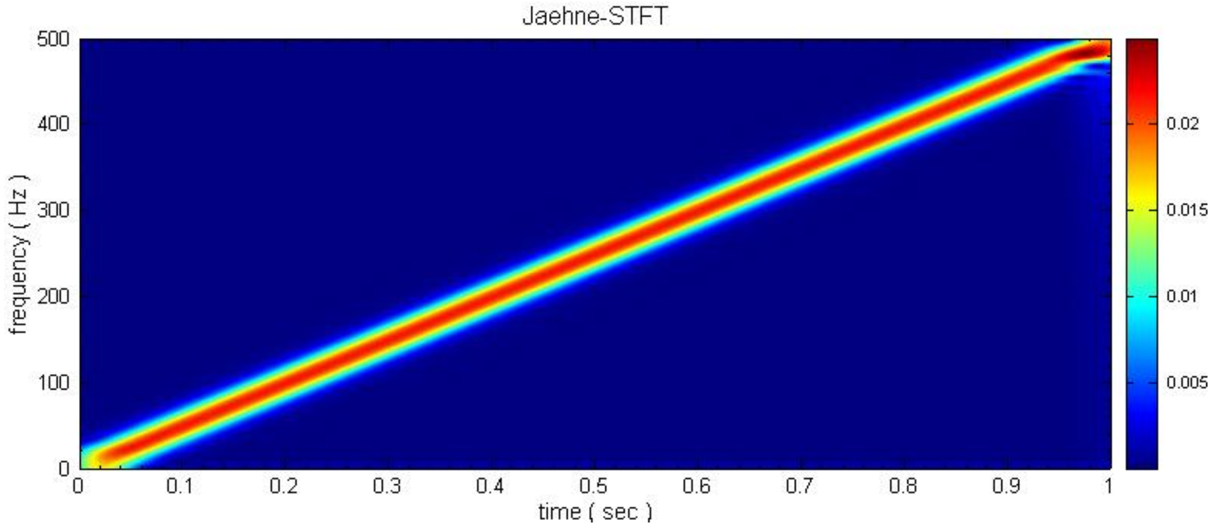
Build the signal with Source / Advanced / Jaehne and viewing its results with Viewer / Channel Viewer.





Connect the Jaehne to Compute / TFA / Short Term Fourier Transform and present the result with Viewer / Time-Frequency Viewer. Jaehne signal is close to Chirp signal. The frequency of the signal varies with time. It increases when time increases.





Related Functions

Channel Viewer, Time-Frequency Viewer, Short Term Fourier Transform.

Chapter 6: Viewer Signal Flow Object

6.1 Annotation (Professional Only)

When data is presented in Channel Viewer or XY Plot, the user can add annotations and curves on the figure.

Introduction

This module includes Ellipses, HLines, HRegions, Lines, Rects, Texts, VLines, VRegions, and other annotation functions.

Properties

Each annotation module is a little bit different in properties. If the coordinate of figures are included in properties, annotations are drawn according to the coordinates in Channel Viewer or XY Plot. They can also be directly added to the proper places using mouse.

Common Properties:

Property Name	Property Definition	Default Value
ZOrder	The layer of the figure. The higher the figure is superposed, the larger the ZOrder is. The original figure is at 0th layer. ZOrder=1 is above the original figure; Zorder=-1 is below the original figure.	-1
LinePen	Plot settings of annotations. The user can set Color, DashStyle, and Width of lines. The properties are separated in Texts.	Color=Red DashStyle=Solid Width = 1

Properties of Ellipses, Lines, Rects (rectangular annotation):

Property Name	Property Definition	Default Value
Start	The starting point of figure’s coordinates. The coordinates are set at the bottom left corner for Ellipses and Rects.	X=0, Y=0
End	The ending point of figure’s coordinates. The coordinates are set at the top right corner for Ellipses and Rects.	X=1, Y=1

Properties of Texts:

Property Name	Property Definition	Default Value
Text	Content of Texts	TEXT
Position	Coordinate of Texts	X=0, Y=0
TextColor	Color of Texts	Red
Textfont	Font of Texts, include size, style, font. It depends on the system installation.	Arial, Size=10

Properties of HRegions (horizontal area) and VRegions (vertical area):

Property Name	Property Definition	Default Value
Position1	The starting coordinate point to plot the figure	0
Position2	The ending coordinate point to plot the figure	1
PixelIndent	The width of the blank border.	0
Color1	Color of the starting point.	Red
Color2	Color of the ending point. The color between Position1 and Position2 is the gradient color from Color1 to Color2 based on the whole figure ratio.	White

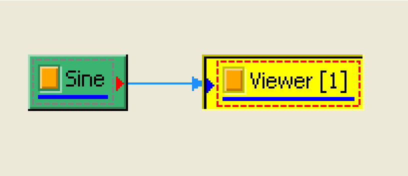
Properties of HLines (horizontal line) and VLines (vertical line):

Property Name	Property Definition	Default Value
Position	The location coordinate point of the line	0
PixelIndent	The width of the blank border.	0

Example

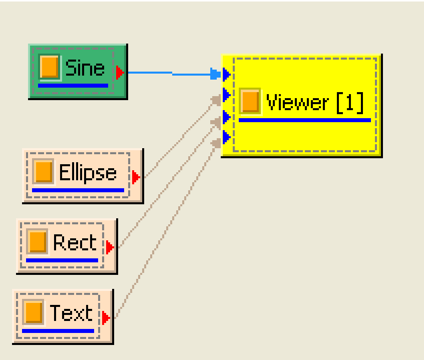
Add annotations for the Sine function.

Create a Sine function with Source / Sine Wave and viewing its results with Viewer / Channel Viewer. Set the ViewerHeight of the Channel Viewer to 400.



Properties	
Appearance	
BackColor	**White**
ViewerWidth	**default (600)**
ViewerHeight	**400**
ListOrder	**0**
RetainPlot	**False**

Add a Rect, a Ellipse, a Text to the Viewer and set the properties as below:



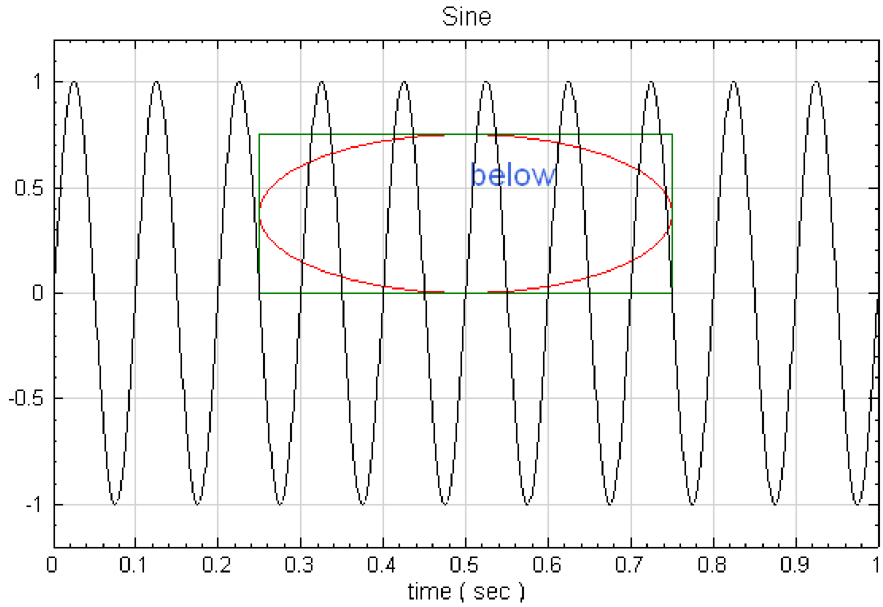
Annotation	
ZOrder	**-1**
Start	**{X=0.25, Y=0}**
End	**{X=0.75, Y=0.75}**
LinePen	**Red, Solid, 1**
Module	

Set the properties of both Ellipse and Rect as above.

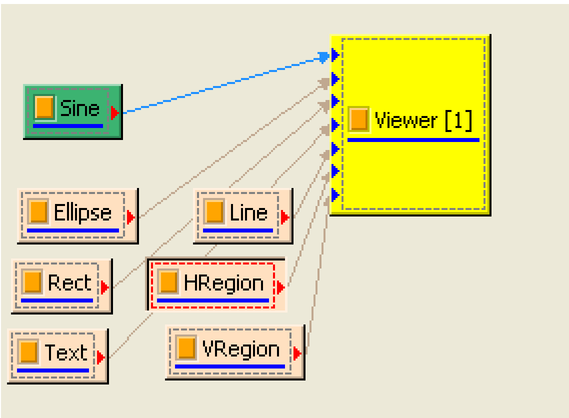
Annotation	
ZOrder	**-1**
Text	**below**
Position	**{X=0.5, Y=0.65}**
TextColor	**RoyalBlue**
TextFont	**Arial, 15.75pt**

Set the properties of the Text as above.

After pressing , we can get the updated figure as below:



Link a Line, a HRegion, a VRegion to the Viewer as shown below and set the properties.



Properties

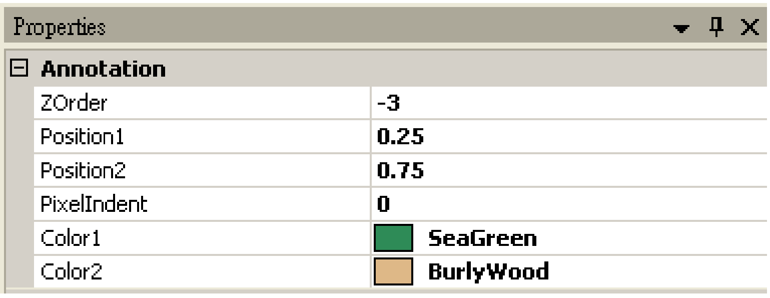
Annotation	
ZOrder	-1
Start	{X=0.2, Y=-0.5}
End	{X=0.4, Y=0.7}
LinePen	DarkGoldenrod, Dot, 5

Set the properties of the Line as above.

Properties

Annotation	
ZOrder	-2
Position1	-0.5
Position2	0.5
PixelIndent	0
Color1	Red
Color2	DarkTurquoise

Set the properties of the HRegion as above.

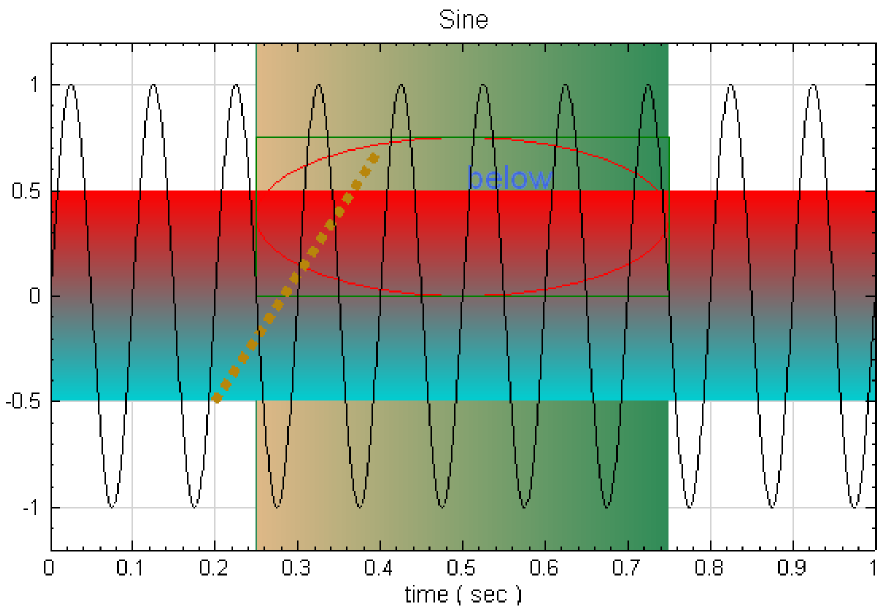


Properties

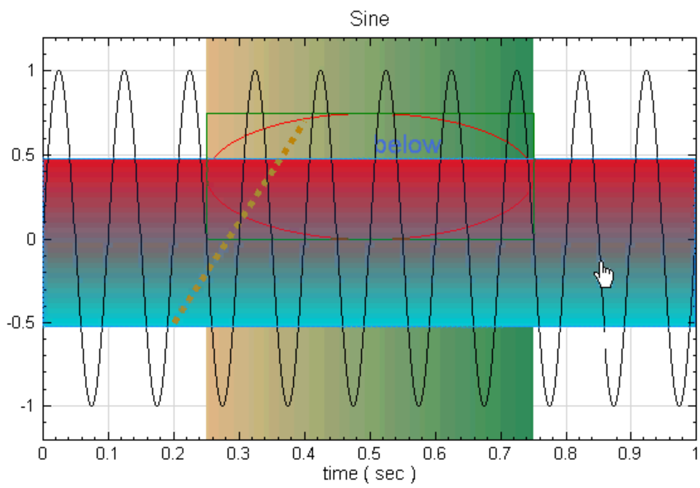
Annotation	
ZOrder	-3
Position1	0.25
Position2	0.75
PixelIndent	0
Color1	SeaGreen
Color2	BurlyWood

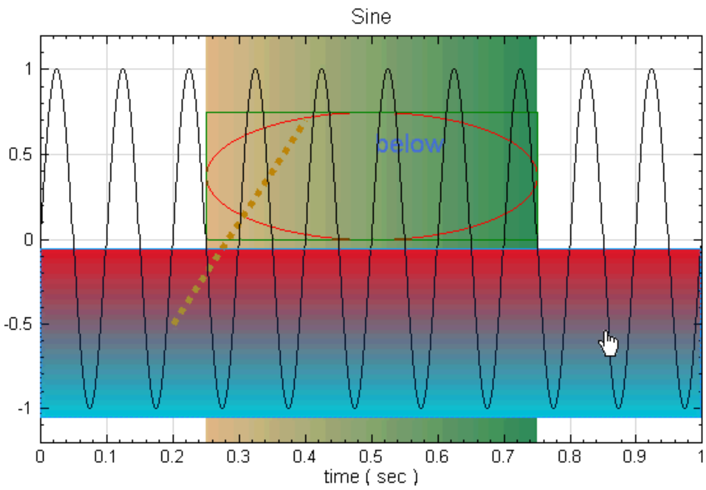
Set the properties of the VRegion as above.

After pressing , we can get the updated figure below:



Move the mouse on the HRegion, the user can drag the HRegion to other locations in the drawing area. Other annotations can be moved in the same way.





Related Functions

Channel Viewer, Sine.

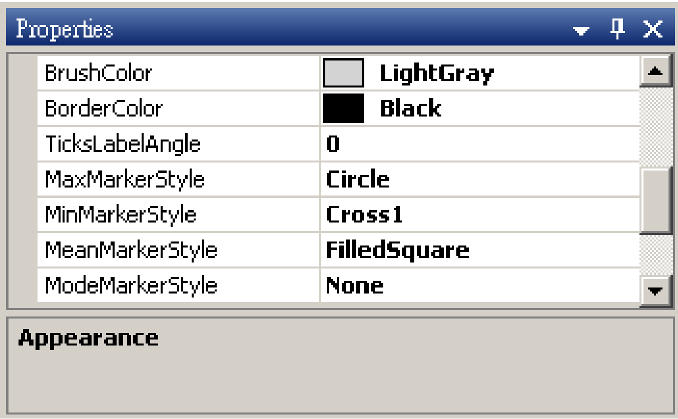
6.2 Box Plot Viewer (Professional Only)

Introduction

Based on the median, the first quartile and the third quartile of a group of numbers, Box Plot Viewer plots boxes. The lines connected with these boxes represent the maximum and minimum value so that statistical property of the data is shown.

Properties

This module accepts real number, complex number, single channel, multi-channel, regular and Indexed signal, or audio input, and supports multi signal input. The user could refer to Channel Viewer for Appearance, Fonts, Colors, Grid and Title parameters. Specific parameters for Box Plot Viewer are introduced below:



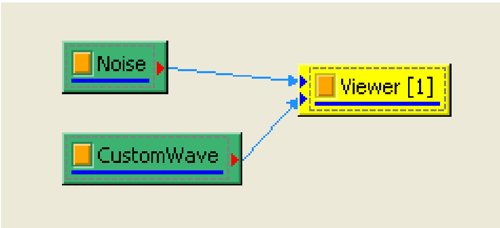
Parameters of Box Plot Viewer

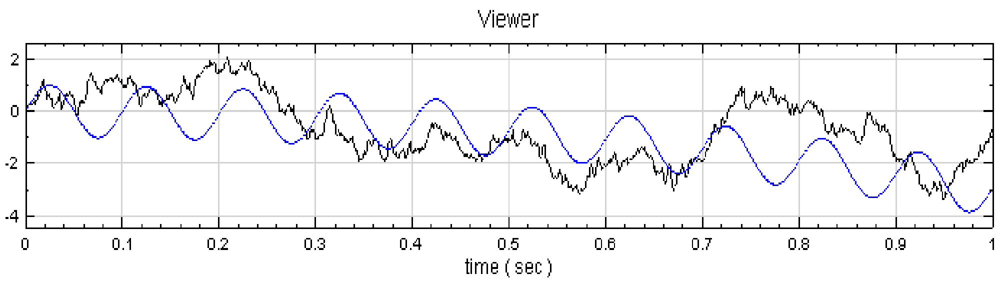
Property Name	Property Definition	Default Value
BrushColor	The brush color inside the Box	LightGrey
BorderColor	The brush color for the border of the Box	Black
TicksLabelAngle	Degree of the angle to arrange the title text below the Box	0
MaxMarkerStyle	The symbol for the maximum value of the series	Circle
MinMarkerStyle	The symbol for the minimum value of the series	Cross1
MeanMarkerStyle	The symbol for the average value of the series	FilledSquare

Example

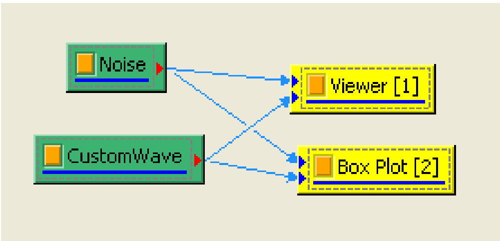
Create a Brownian Noise and a CustomWave, signals are plotted by Channel Viewer, then box figures of these two signals are drawn via Box Plot Viewer.

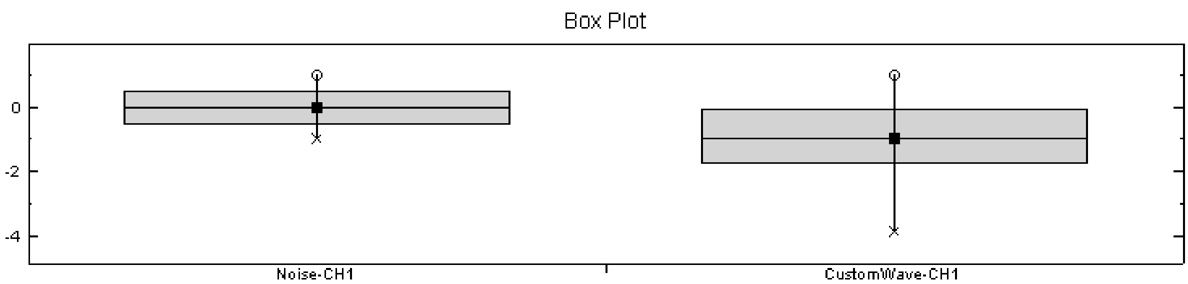
Generate a noise signal by Source / Noise and adjuste the property of the noise to Brown via Properties / NoiseType. And create a CustomWave using Source / Custom Wave and set the equation “sin (2 * pi * 10* t) - 3 * pow (t , 2)” in Properties / Expression. Then these two signals are connected to Channel Viewer.





Connect both signals to Viewer / Box Plot Viewer, the box view is shown below.





Related functions

TF Viewer, interface, Map to Real.

6.3 Channel Viewer

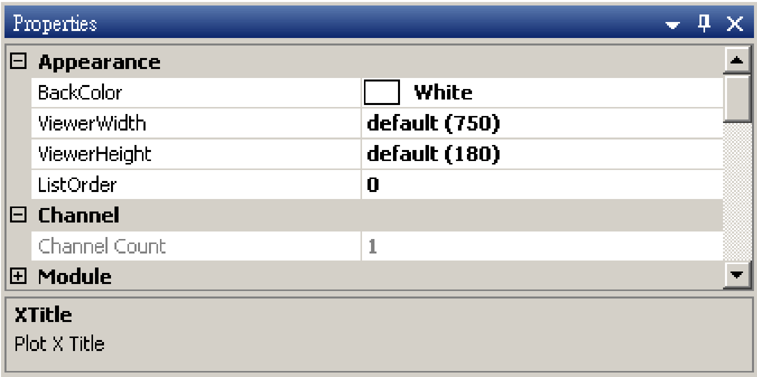
The purpose of the Channel Viewer is to convert signal data to graphical display onto the *Visualization Window*. The graph will plot each signal data along the x-axis (time).

Properties

This module accepts input of Signal (which can be real number or complex number, single channel or multi-channel, Regular or Indexed), Audio (which can be real number or complex number, single channel or multi-channel, Regular). Chanel Viewer can accept multiple input data sources.

1. Appearance

The Appearance property contains the options to set the appearance of how the graph of the Channel Viewer will be shown on the *Visualization Window*.



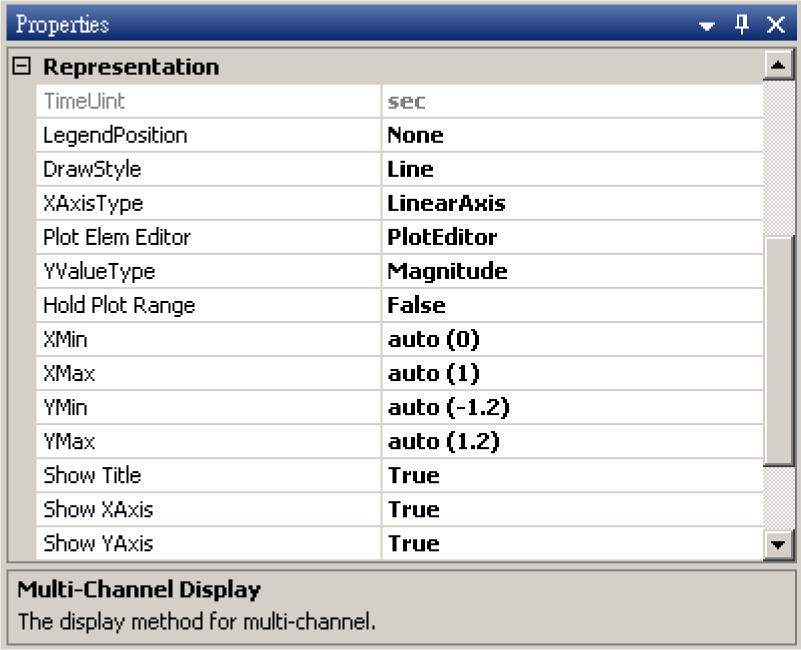
Property Name	Property Definition	Default Value
BackColor	Set the background color of the graph displayed in the *Visualization Window*.	White
ViewerWidth	Set the width of the graph in pixels.	650
ViewerHeight	Set the height of the graph in pixels.	180

ListOrder	Set the order of the graph to be shown on the *Visualization Window*.	The default position on the *Visualization Window* is based on the order that the Channel Viewer is created.

2. Channel

Property Name	Property Definition	Default Value
Channel Count	Displays the number of input signals currently connected to the Channel Viewer.	(Cannot be edited)
Multi-Channel Display	Select from the option Overlapped (to display the graphs of the input signals on the same graph overlapping each other) or List (to display the graphs on top of one another).	Overlapped
Show value Channel	When there are multiple inputs, select the channel (graph) from the drop down menu to use the Show Value button on the *Visualization Window Toolbar*. When there are multiple inputs, knowing which graph shows what value can be rather difficult. So selecting a channel from the drop down menu, the user can specify the graph to perform the Show Value button (located on the *Visualization Window Toolbar*).	Channel 1

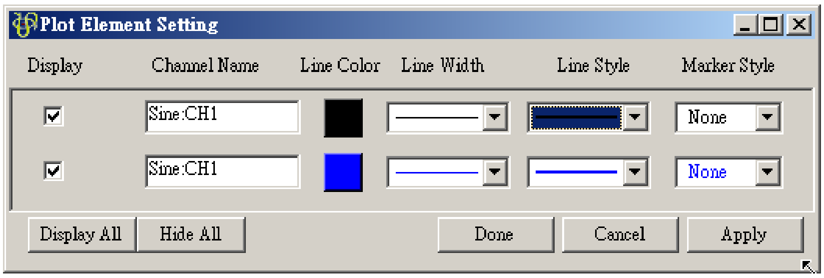
3. Representation



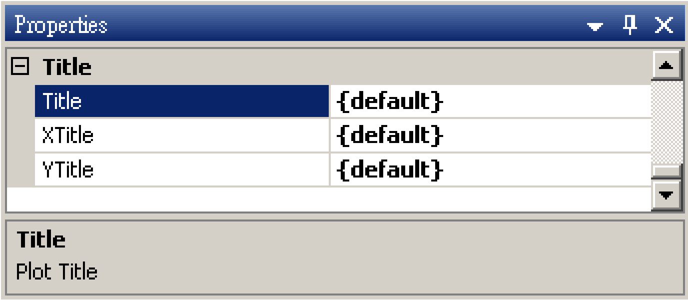
Property Name	**Property Definition**	**Variable Value**
TimeUnit	Displays the time unit of the data.	Depends on the input signal data’s time unit.
LegendPosition	Select the position: None, TopLeft, BottomLeft, TopRight, BottomRight and RightOutSide to display the legend on the graph.	None
DrawStyle	Select between Line and Steam to determine how the graph is drawn.	Line
XAxisType	Select the representation of the x-axis, choose between Linear Axis and Log Axis.	LinearAxis
PlotElemEditor	Click on the Plot Editor button next to the field to edit how the graph is displayed, from the line color, line thickness, dot representation etc (**Note:** User will need to click on the PlotElemEditor field for the button to appear).	default

YValueType	Select different ways to display the y-axis from a selection of Magnitude, Phase, Real, Imagine, Gain and Powerspectrum. Normally this option is used for spectrum data. When there are multiple channels in the signal. (**TIPS:** Please look up Chapter 4.4 Map to Real for more information).	Magnitude
GainReference	Conversion→Map to Real. If YValueType is set as Gain, this option field will appear. (**TIPS:** Please look up Chapter 4.4 Map to Real for more information).	1
HoldPlotRange	When HoldPlotRange is set as True, after resizing, moving and zooming into the graph, the calculation done will still be based on the original range.	False
Xmin	Set the minimum value of the x-axis.	auto
Xmax	Set the maximum value of the x-axis.	auto
Ymin	Set the minimum value of the y-axis.	auto
Ymax	Set the maximum value of the y-axis.	auto
Show Title	Select True to show the title on the graph and False to hide the title.	True
Show X Axis	Select True to show the x-axis on the graph and False to hide the x-axis.	True
Show Y Axis	Select True to show the y-axis on the graph and False to hide the y-axis.	True

Clicking on the *PlotElemEditor* button will pop up the *Plot Element Setting* window. Check the *Display* tick box to show the signal on the graph (useful to determine which is which when there are multiple signals on one graph). You can change the *Channel Name*, *Line Color*, *Line Width*, *Line Style* and *Marker Style* to improve the presentation of the graph and customize the looks according to your need.



4. Title

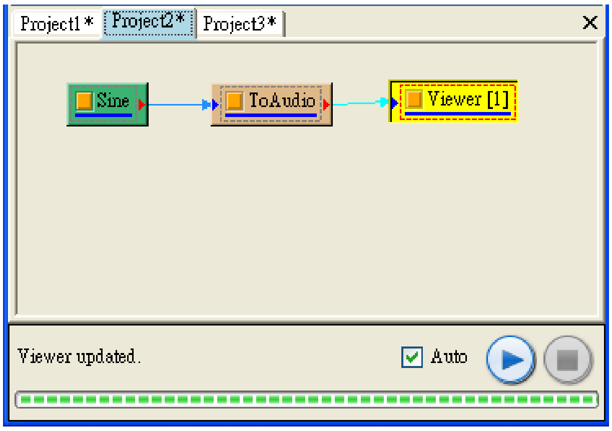


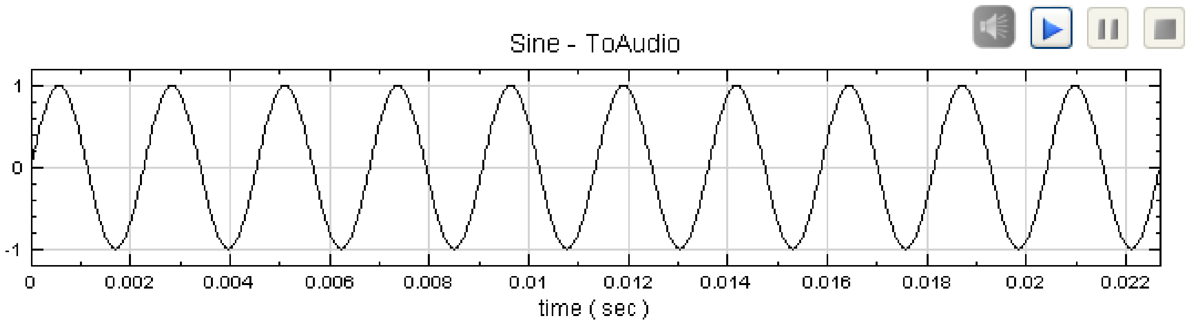
Property Name	Property Definition	Default Value
Title	Change the title of the graph.	Default name of the input SFO.
Xtitle	Change the title of the x-axis.	time
Ytitle	Change the title of the y-axis.	None

Example

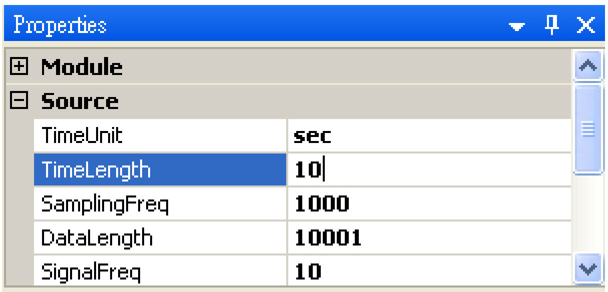
A demonstration of how to use the Audio Player.

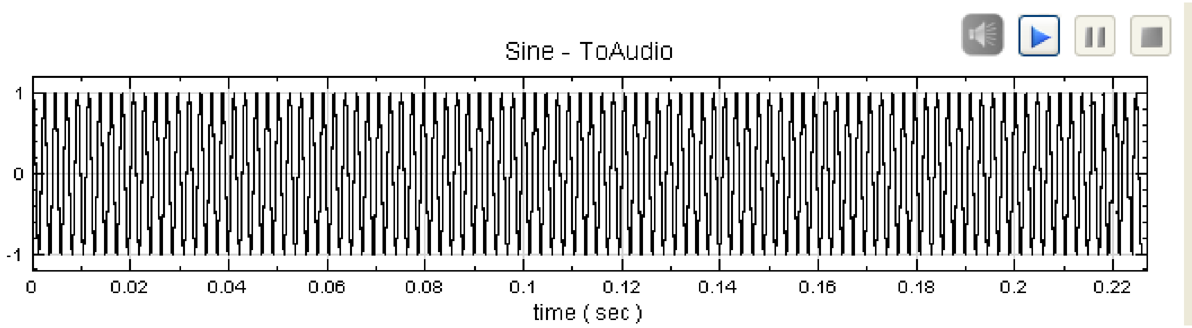
1. Create *Source*→*Sine Wave* and connect the Sine Wave SFO to *Conversion*→ *Converter To Audio* to turn the Sine Wave signal to an Audio file. Then connect the Converter to Audio SFO to *Viewer*→*Channel* to display the graph and the audio component on the *Visualization Window*.



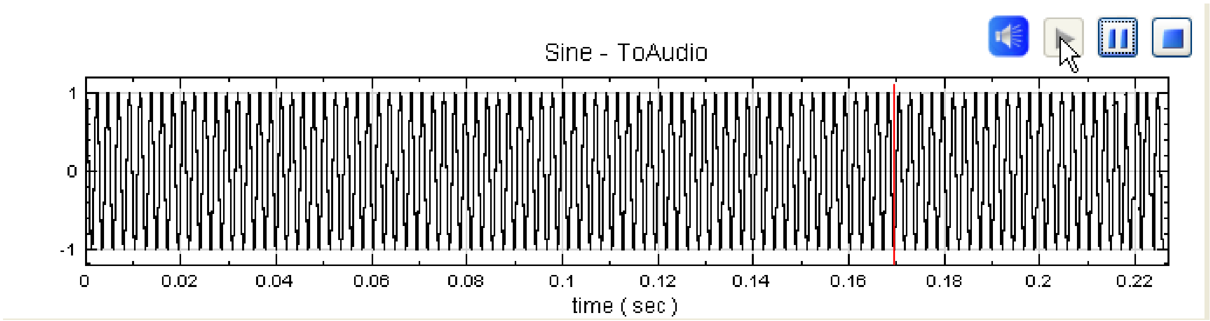


2. Change the *Properties/Source/TimeLength* of the Sine Wave SFO to 10.

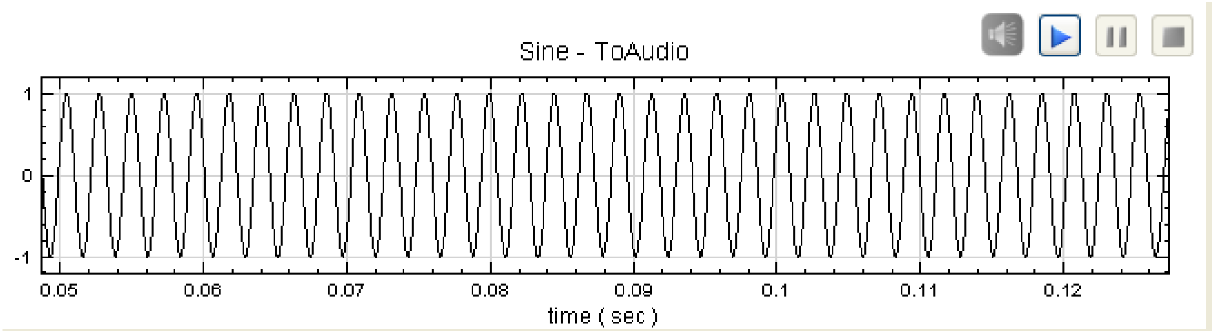




3. Click on the  audio play button on the top right corner of the graph and play the signal. A red line will run through the x-axis indicating the position of the audio currently being played.

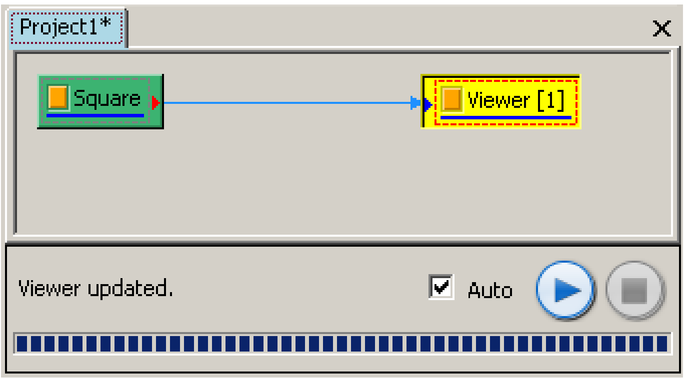


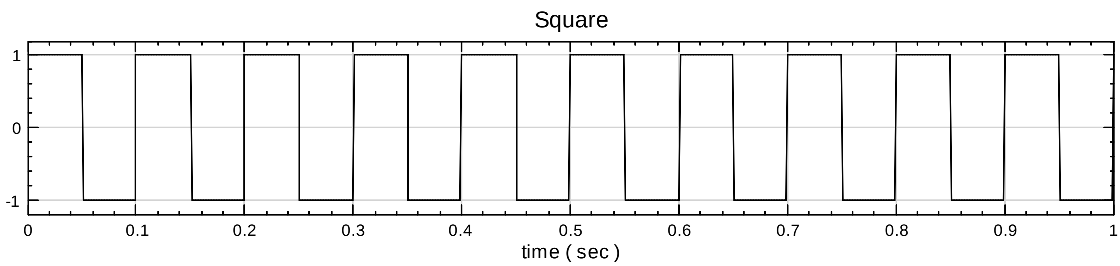
4. You can also use the *Zoom X* button off the *Visualization Window Toolbar* to enlarge the area of the Audio signal.



Below are some examples showing how to configure the other options in the *Properties Window*.

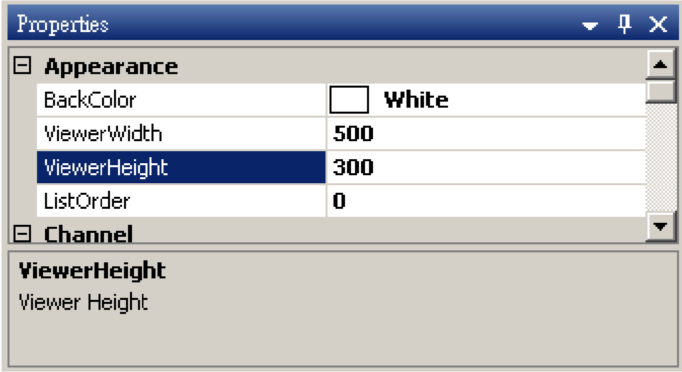
1. Create *Source→Square Wave* and connect it to *Viewer→Channel Viewer*.

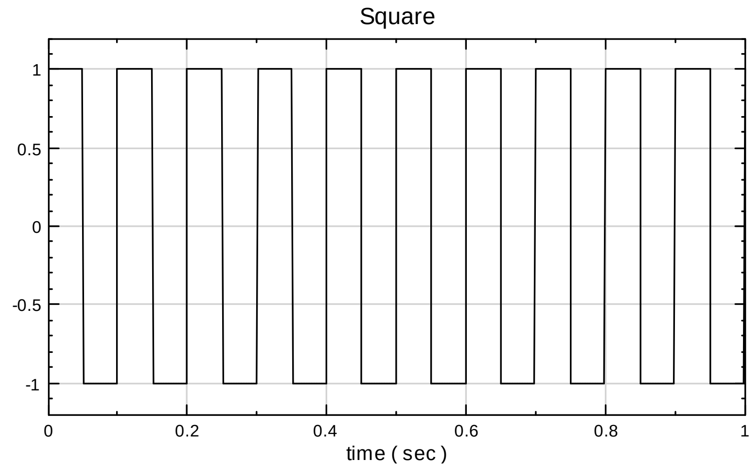




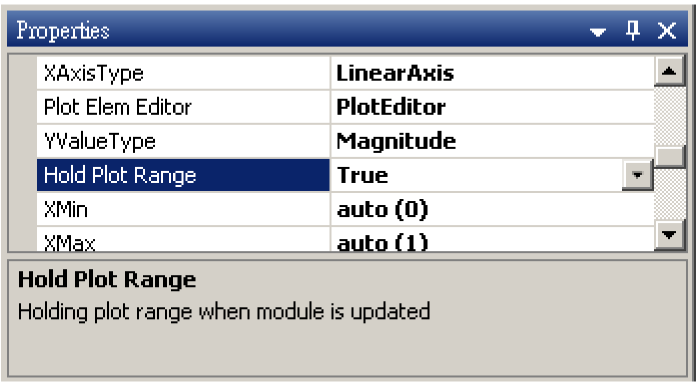
2. Change the Channel Viewer's *Properties/Appearance/ViewerHeight* to 500 and

ViewerWidth to 300.

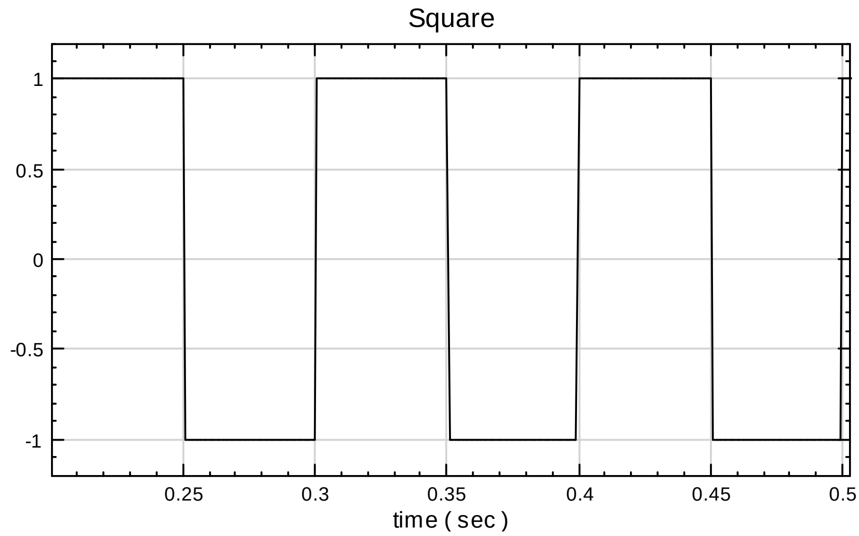




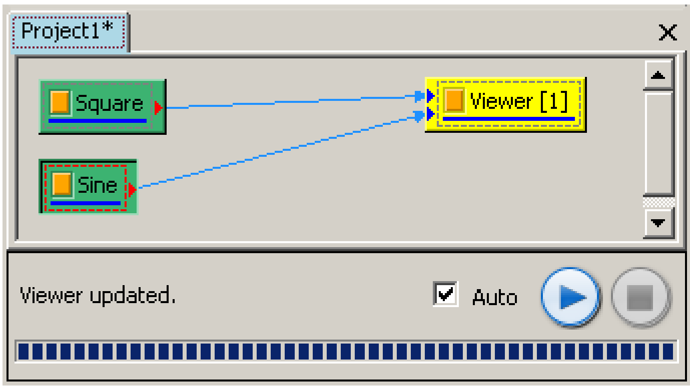
3. Use the *Zoom X*, *Zoom Y* or *Pan X* and *Pan Y* feature of the *Visualization Window Toolbar*. If you want to maintain the current status, you can set the *Properties/Representation/HoldPlotRange* to *True*.

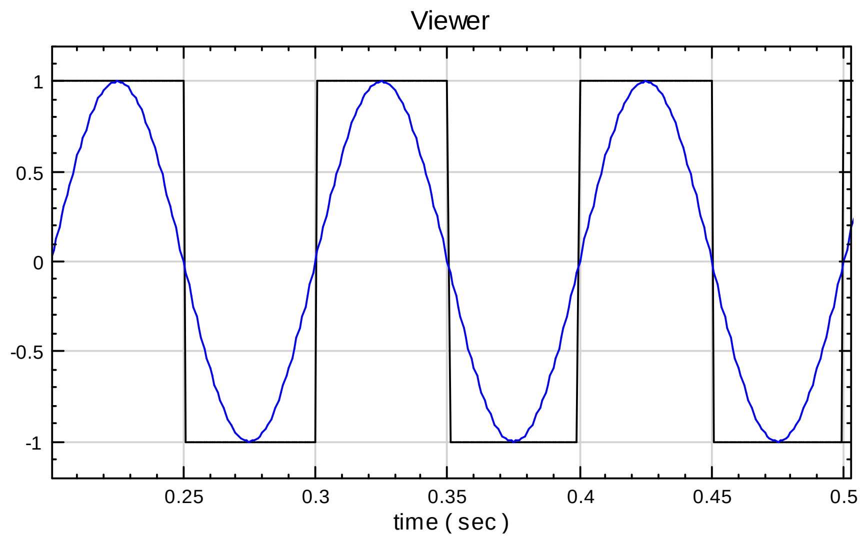


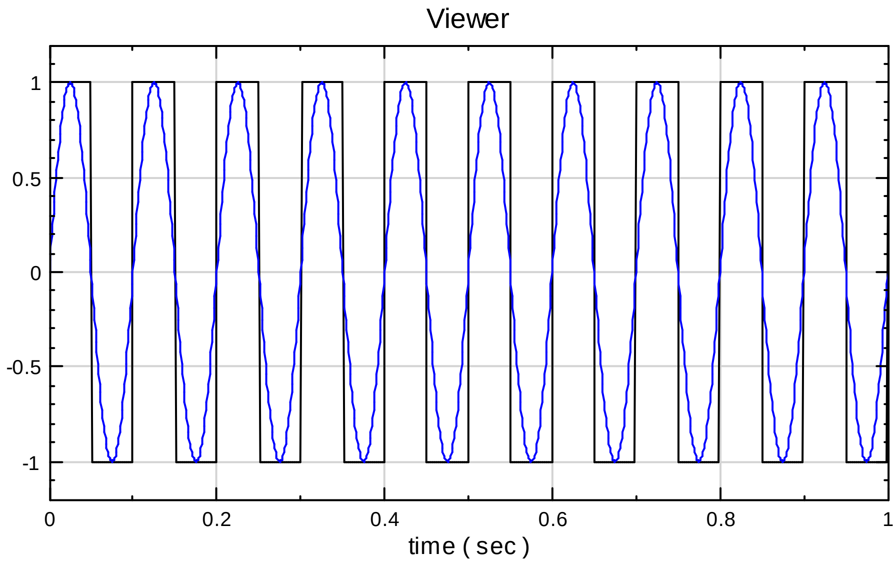
4. Use *Zoom X* and zoom in-between 0.2 sec and 0.5 sec on the x-axis.



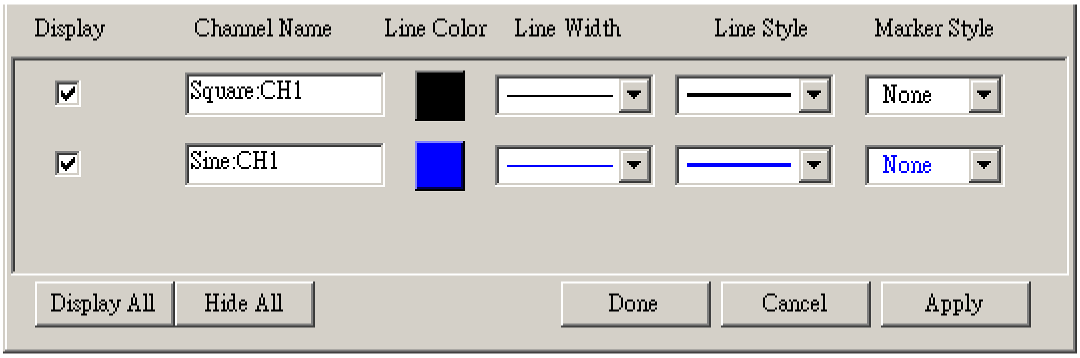
5. Create *Source→Sine Wave* and connect it to the same Channel Viewer SFO. Because the Auto box is checked, the Channel Viewer automatically updates with the new Sine wave graph. Since *HoldPlotRange* is set to *True*, the new update does not return the graph to the default position.



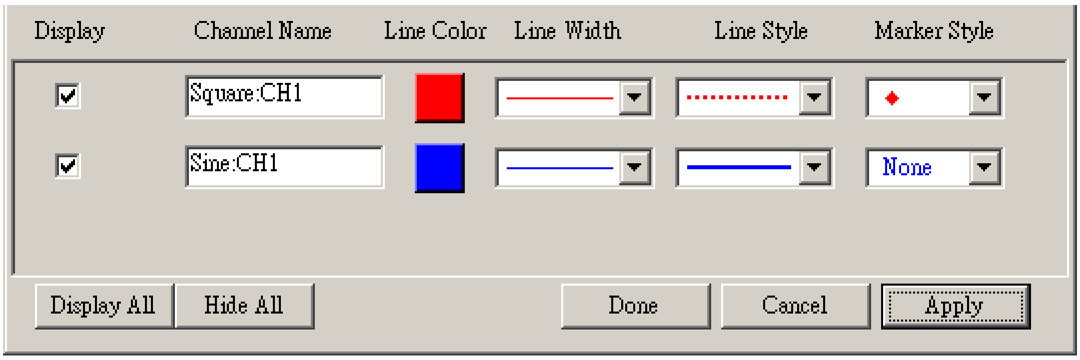


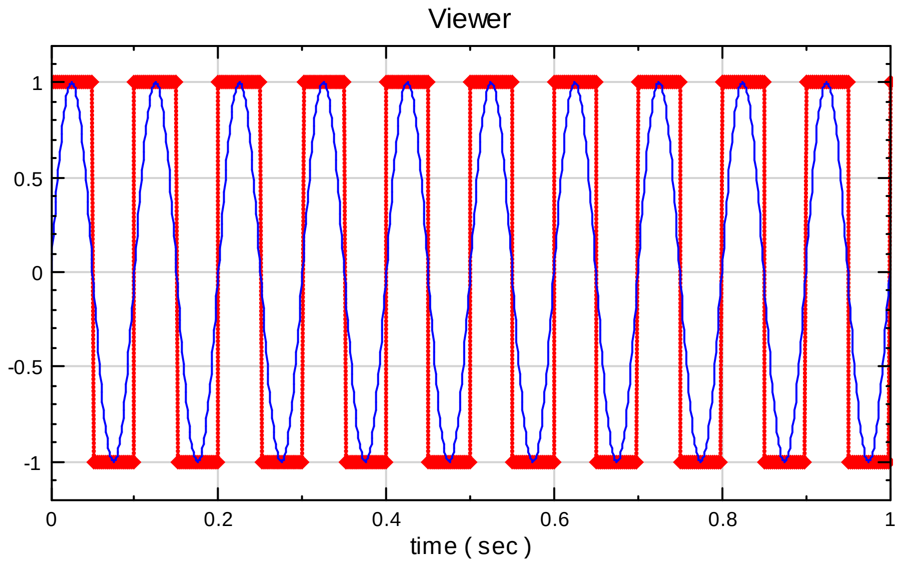


6. Click on the *Properties/Representation/Plot Elem Editor* and click on the button to open up the *Plot Element Setting* window.

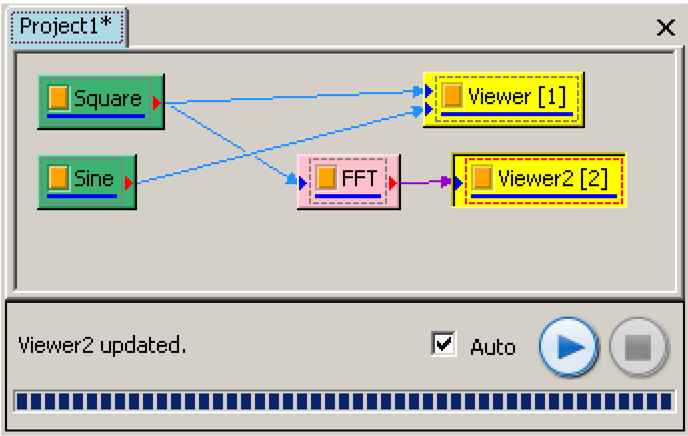


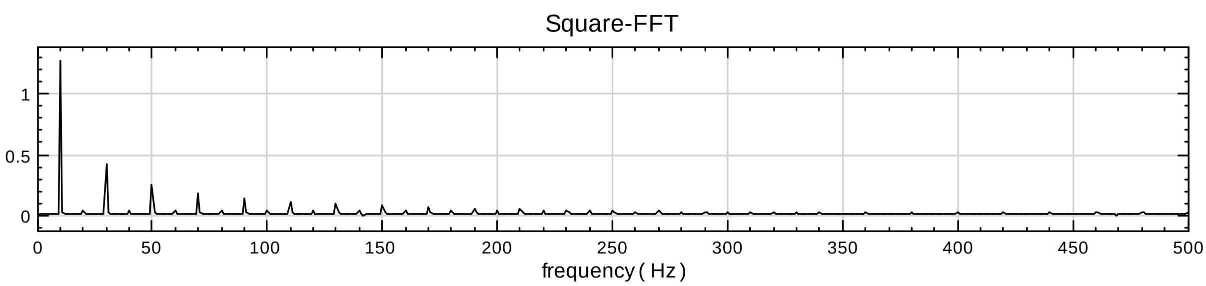
In the Plot Element Setting window, you can edit the display of all the input channel data on the graph. Set the Line Color of Square:CH1 to red, change the Line Style to dotted line and change the Marker to ◆ and click on the Apply button to see the change on the graph.



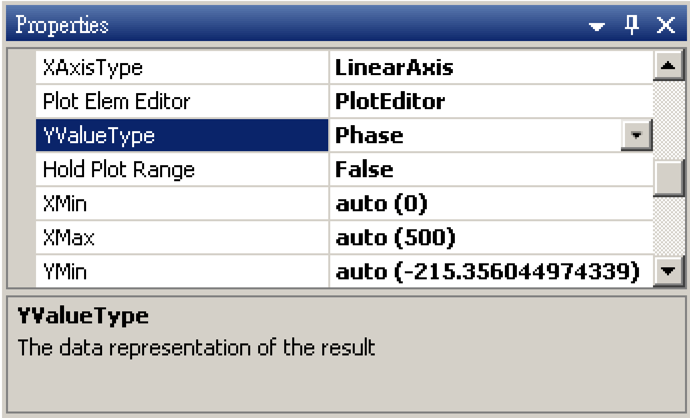


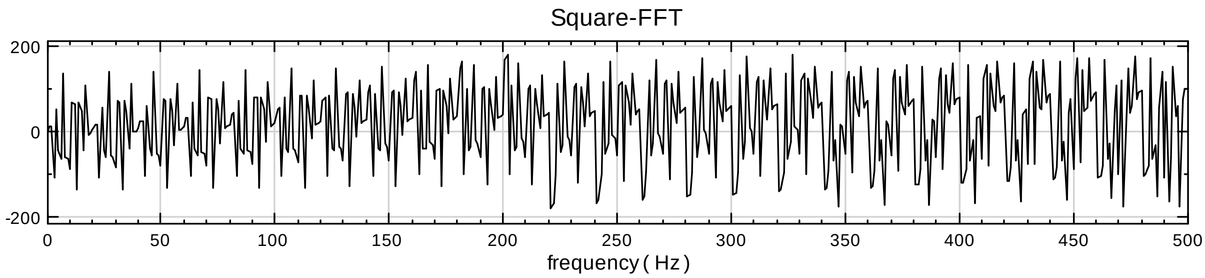
7. Configuring the YValueType option of the Channel Viewer on a spectrum data such as magnitude, phase etc. Continuing from the above example, connect the Square Wave SFO to *Compute→Transform→Fourier Transform (FFT*) and then connect it to a Channel Viewer.



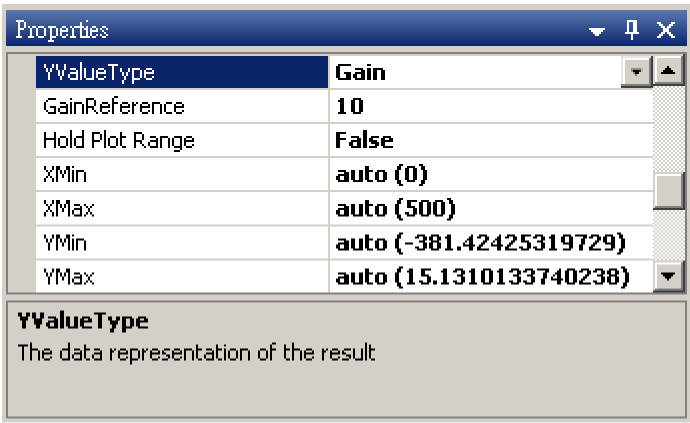


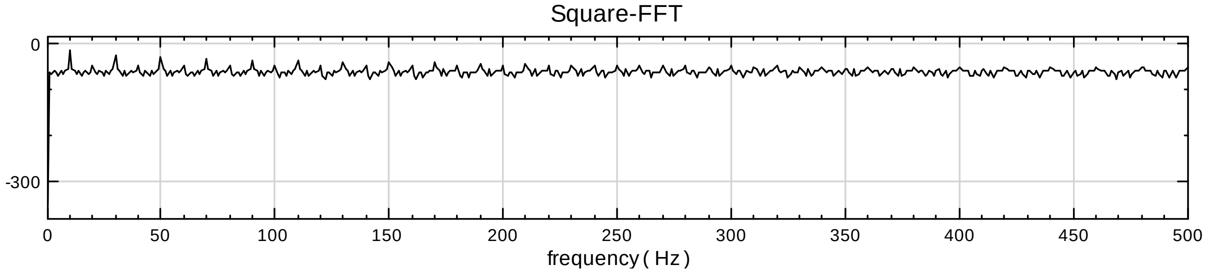
In the graph, the x-axis is the frequency and the y-axis YValueType is set as Magnitude. Now change YValueType to Phase, the y-axis represents the frequency of each phase.





When YValueType is changed to Gain, an additional option GainReference will appear. Gain is defined as $20\log(\frac{A}{\mathrm{GainRef}})$, unit is dB, log is to the base of 10, *A* is the Magnitude and the denominator is GainReference.





Related Functions

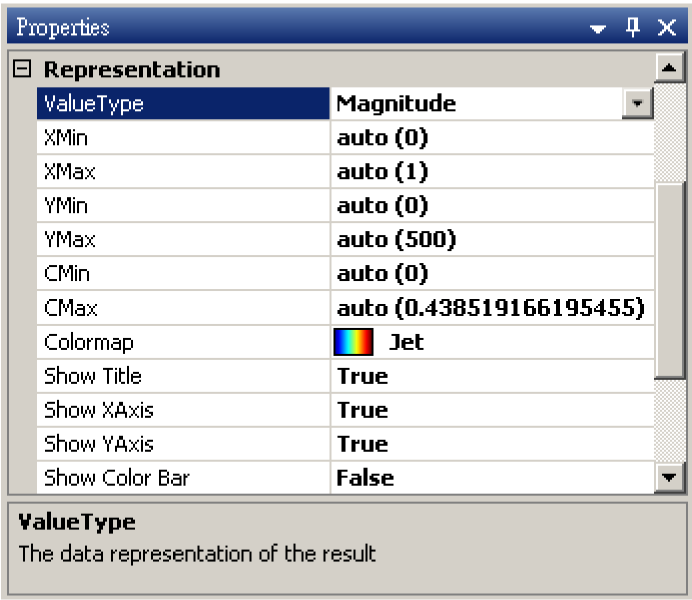
TF Viewer, User Interface, Map to Real.

6.4 Time-Frequency Viewer

Time-Frequency Viewer uses images to display three dimensional time-frequency signals (time, frequency and signal strength). The x-axis represents the time, the y-axis represents the frequency and the color represents the signal strength.

Properties

This module accepts input of Spectra (which could be real number or complex number, single channel, Regular). Time-frequency Viewer and Channel Viewer are very similar with the difference being that there are more variable options for Time-frequency Viewer.

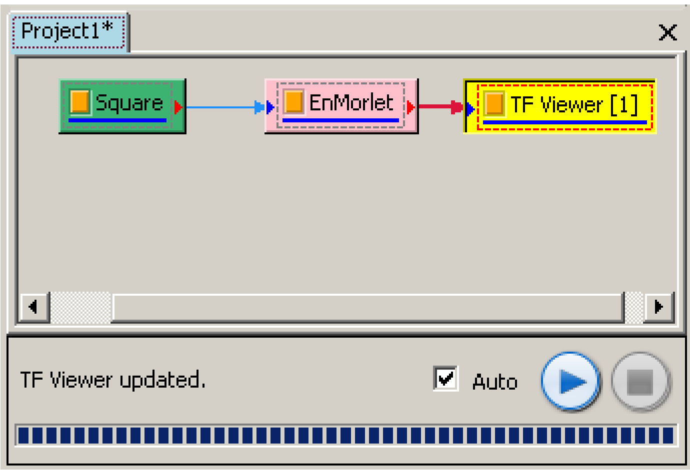


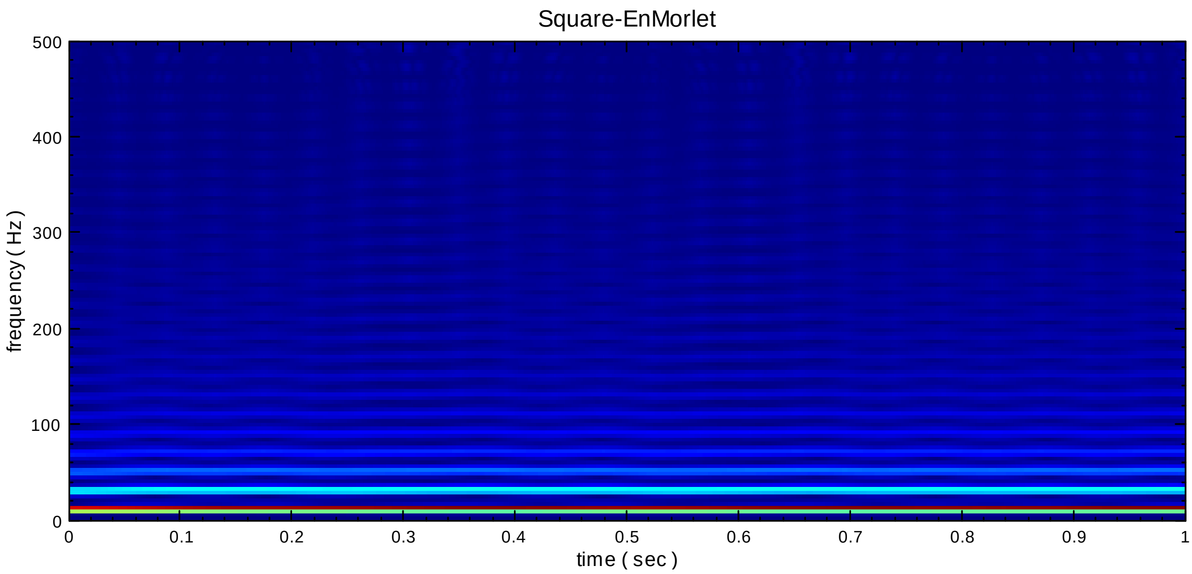
Property Name	Property Definition	Default Value
CMin	Set the minimum value of the time-frequency color.	Auto
CMax	Set the maximum value of the time-frequency color.	Auto
Colormap	There are four types of color representation: Jet, Hsv, Rainbow and Gray.	Jet
Show Color Bar	Select whether or not to display the color bar at the right side of the graph.	False

Example:

Create a Square Wave SFO and connect it to an Enhanced Morlet Transform SFO and then connect it to a Time-frequency Viewer SFO. Then change some configuration to the Time-Frequency Viewer.

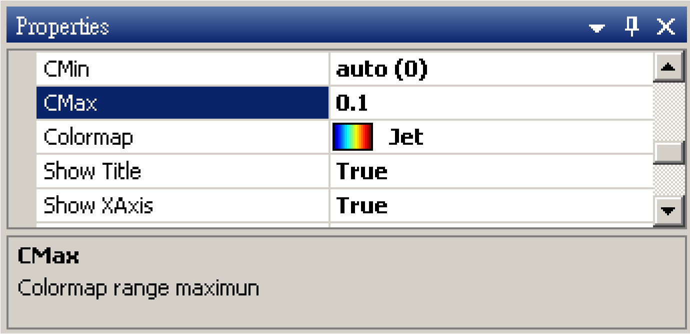
1. Create *Source→Square Wave* and connect it to *Compute→TFA→Enhanced Morlet Transform* and then connect it to a Time-Frequency Viewer SFO. Select the ValueType as *Properties/Representation/Magnitude*.

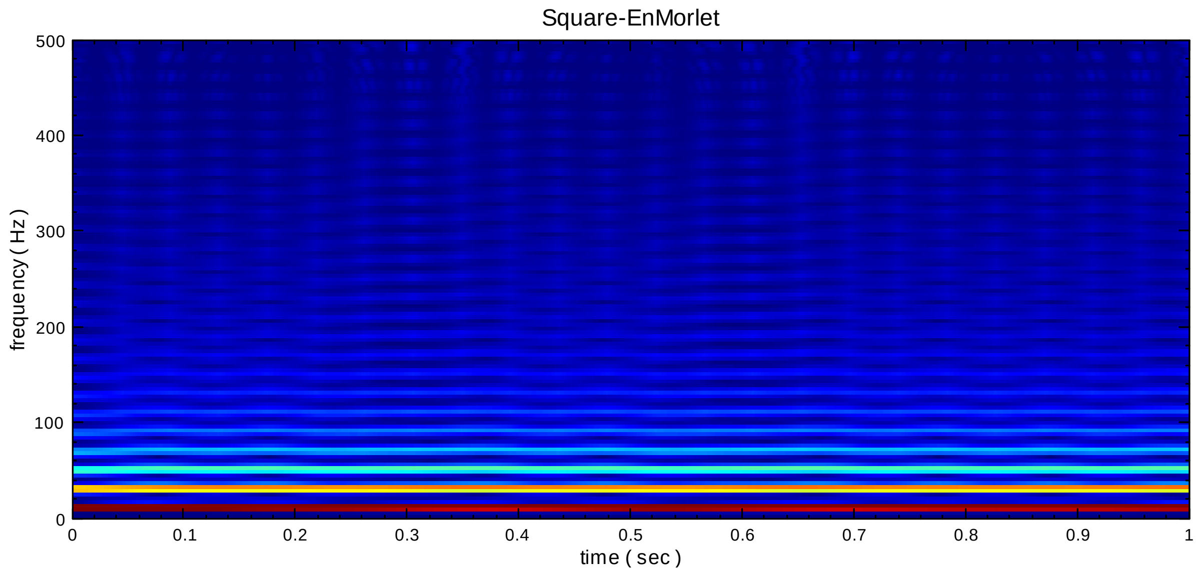




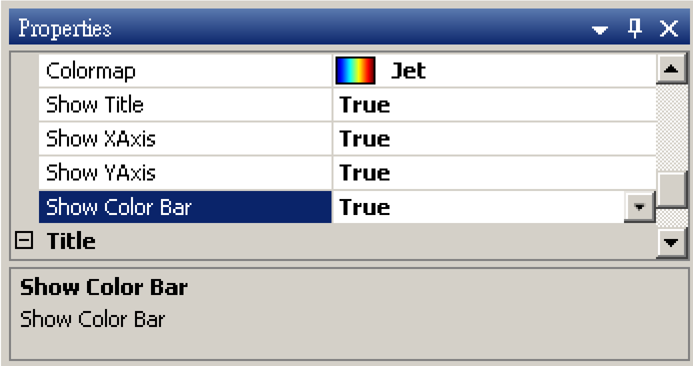
Set the *Properties/Representation/CMax* to 0.197772704820884. This value is the maximum value of the signal strength and it is also the maximum color value on the Colormap. A user can set the value of the CMax variable to show the signal strength below this value. Since the colors on the Colormap keep the same, a better resolution of the signal strength can be presented if CMax becomes smaller.

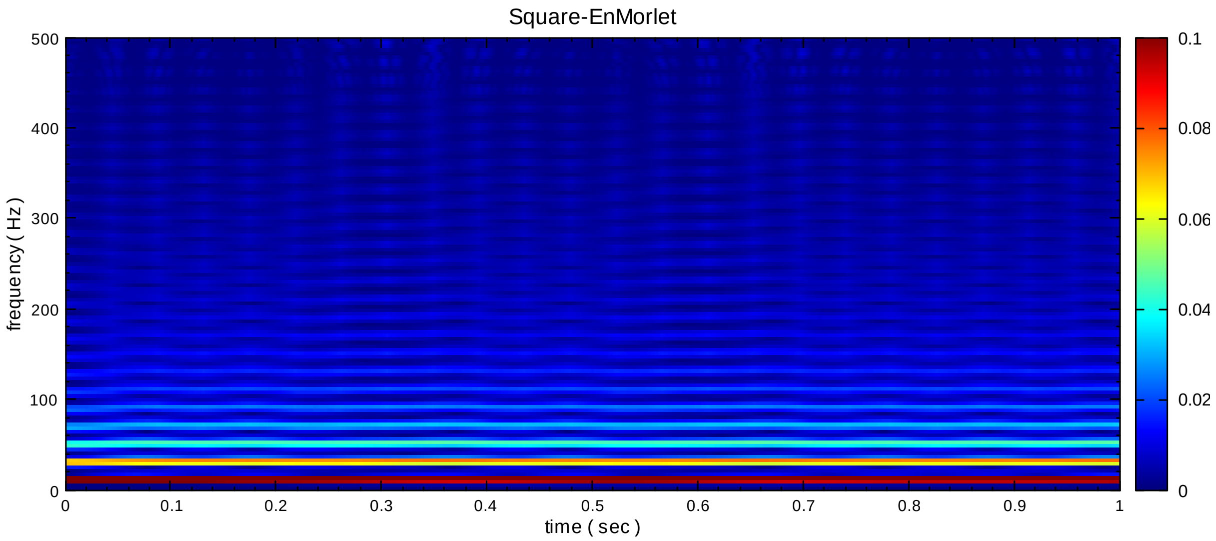
Set CMax to 0.1, the graph uses this as the maximum color value for Colormap. All signal strength below 0.1 is remapped to the Colormap and the graph is redrawn to focus on the region that was unclear when CMax was 0.197772704820884.



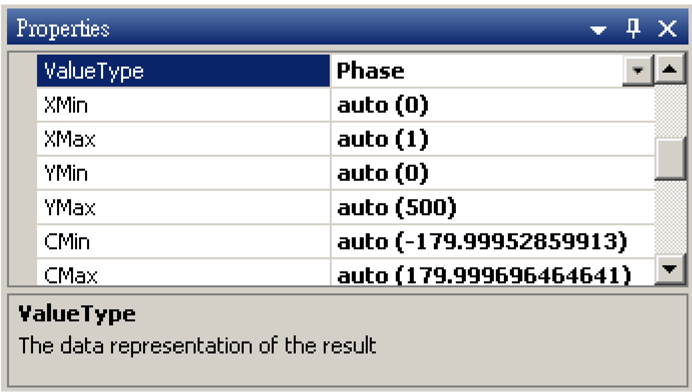


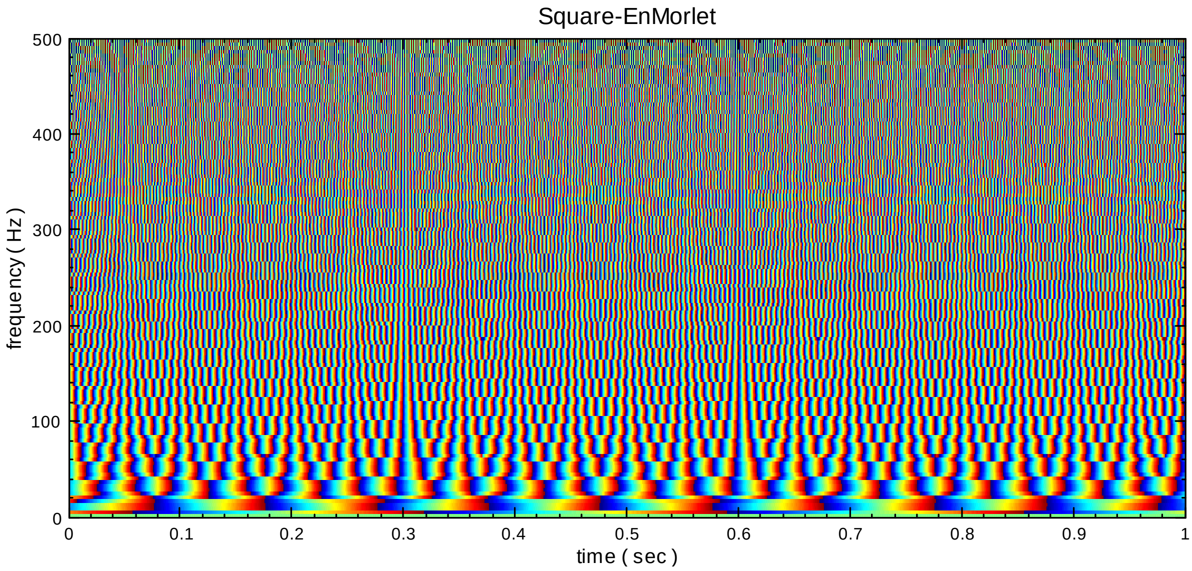
2. Set the *Show Color Bar* to *True* will display a color bar legend on relationship between the color and its values.





3. Change the Time-Frequency Viewer's *Properties/Representation/ValueType* to *Phase* and the following image will be displayed.





Related Functions

Channel Viewer, User Interface, Map to Real.

6.5 XY Plot Viewer

Display two signal data, one corresponding to the x-axis and the other corresponding to the y-axis.

Introduction

XY Plot Viewer accepts three main signal data:

1. Two signal data, Channel 1 is drawn on the x-axis and Channel 2 is drawn on the y-axis and then the two signals are plotted on the graph.

2. Multi-Channel data with odd Channels are drawn on the x-axis and even Channels are drawn on the y-axis and then the channels are plotted on the graph.

3. A single channel with multiple data, real part is drawn on the x-axis and the imaginary part is drawn on the y-axis and the two values are plotted on the graph.

Properties

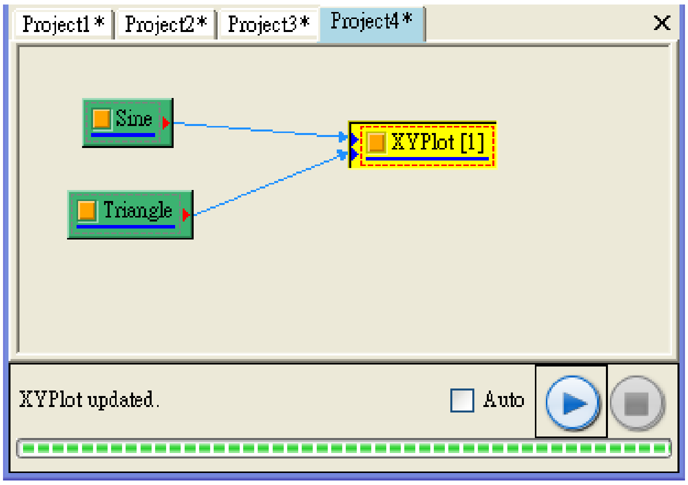
This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular or Indexed), Audio (which could be real number or complex number, single channel or multi-channel, Regular). XY Plot Viewer and Channel Viewer are very similar with the difference being that there are more variable options for XY Plot Viewer.

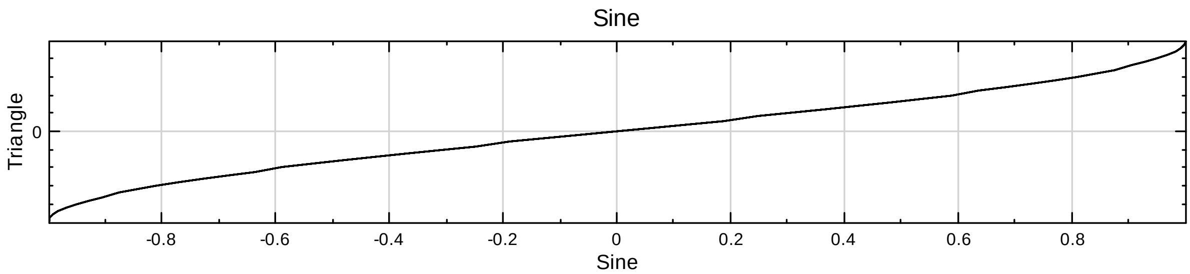
Property Name	Property Definition	Default Value
MaxPointCount	The number of points to be drawn.	1001

Example:

Example 1: Sine Wave is drawn on the axis and Triangle Wave is drawn on the y-axis and then use the XY Plot Viewer to display the graph.

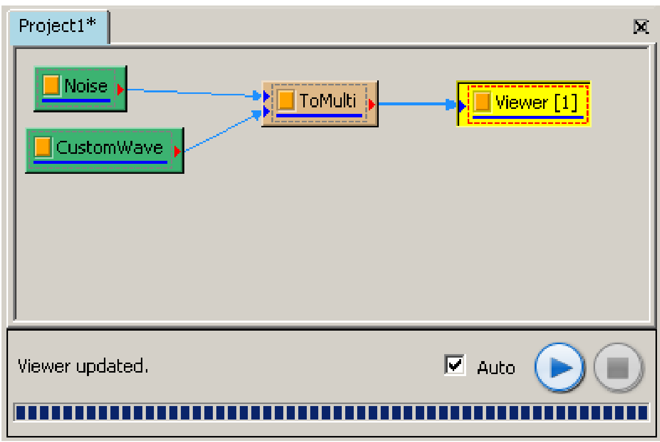
1. Create *Source→Sine Wave* and then create *Source→Triangle Wave* and connect both signal data to *Viewer→XY Plot Viewer*.

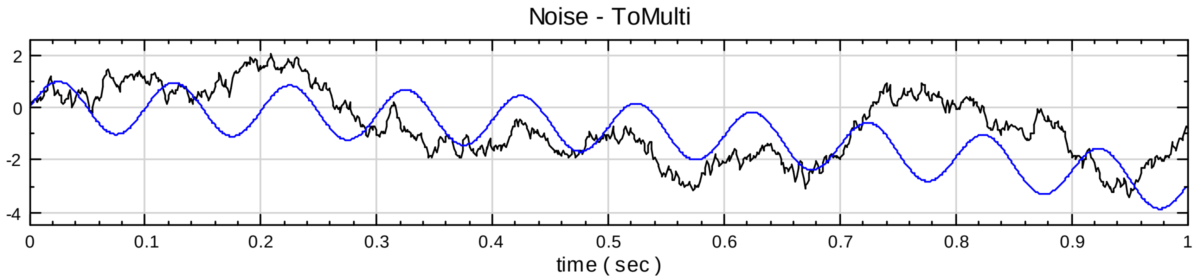




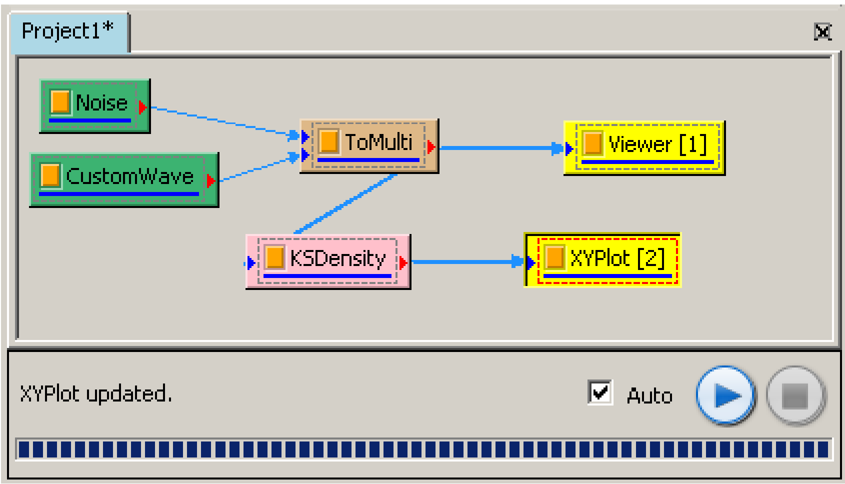
Example 2: Create a Brownian Noise and a CustomWave and calculate them using Kernel Smoothing Density. Now connect both signal data to a XY Plot Viewer to display the graph.

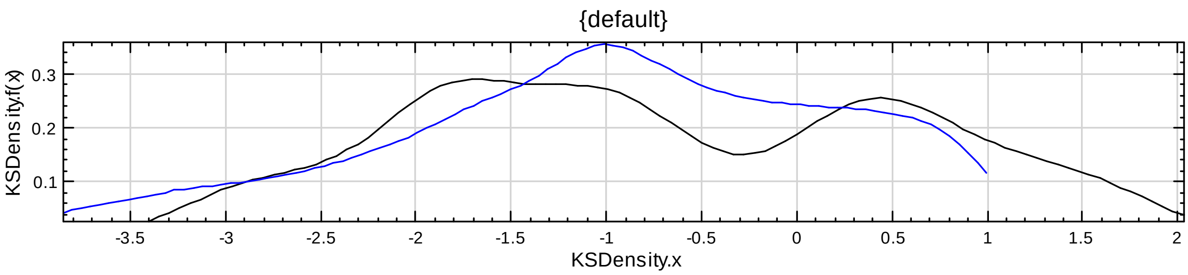
1. Create *Source→Noise* and change its *Properties/Noise/NoiseType* to Brown. Then create *Source→Custom Wave* and set its *Properties/Source/Expression* to **sin(2*pi*10*t)-3*t*t**. Connect both signal data to a *Conversion→Merge to Multi-Channel* and then connect it to *Viewer→Channel Viewer* to display the data.





2. Then connect the ToMulti SFO to a *Compute→Statistics→Kernel Smoothing Density* and then connect it to a *XY Plot Viewer*.

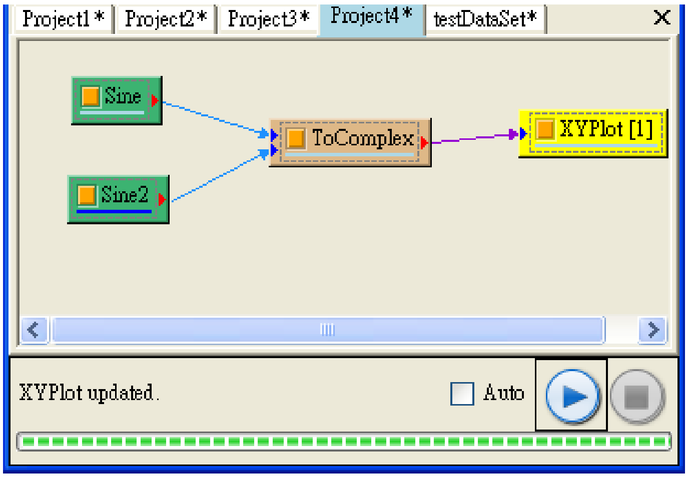


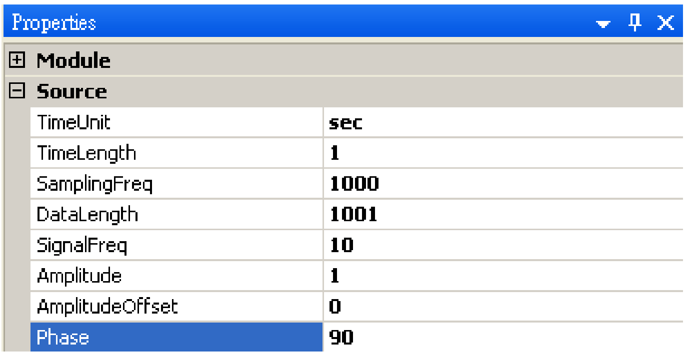


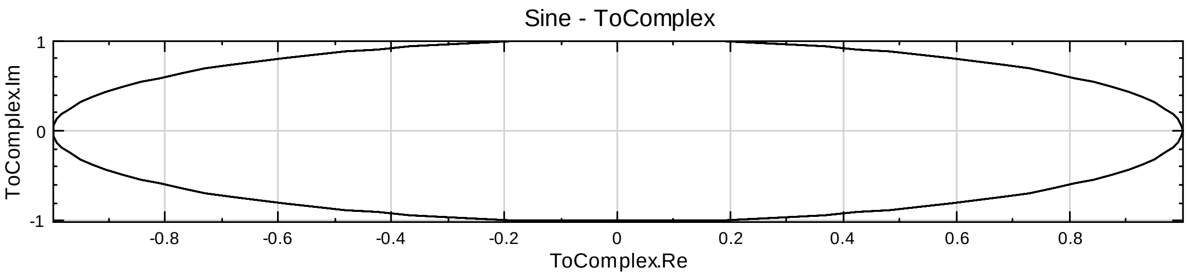
Example 3: The x-axis on the XY Plot represents the odd numbers of the Channel (signal information) and the y-axis on the XY Plot represents the even number of the Channel (probability density function).

A single channel multi-data signal will be drawn by the XY Plot Viewer.

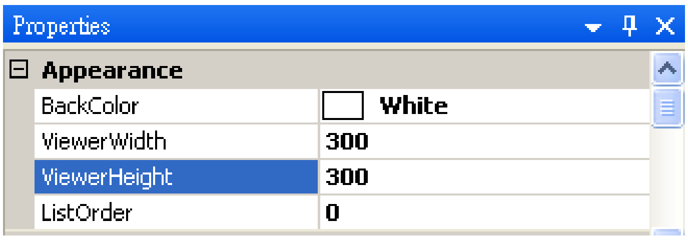
1. Create two *Source→Sine Wave* signals and change the second Sine Wave SFO's *Properties/Source/Phase* to 90 (which will create a Cosine Wave). Connect both signal waves to *Conversion→To Complex* to combine the signal data to a single channel multi-data signal and then connect it to a XY Plot Viewer to display the graph.

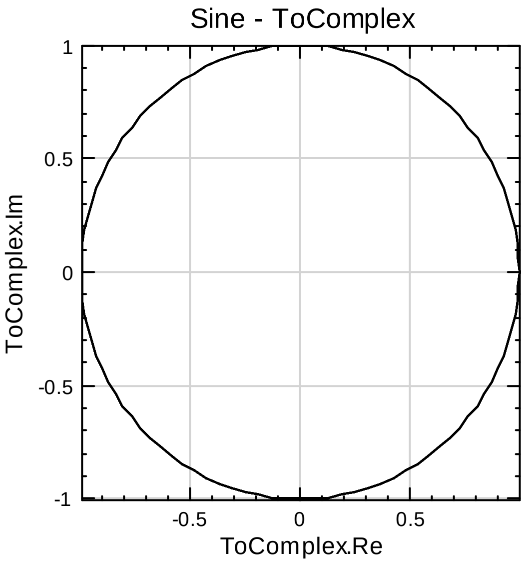






2. Set both the *Properties/Appearance/Viewer Width* and *Viewer Height* to 300 and have the XY Plot Viewer update the graph.





Related Functions

Channel Viewer, Kernel Smoothing Density.

6.6 Histogram Viewer (Professional Only)

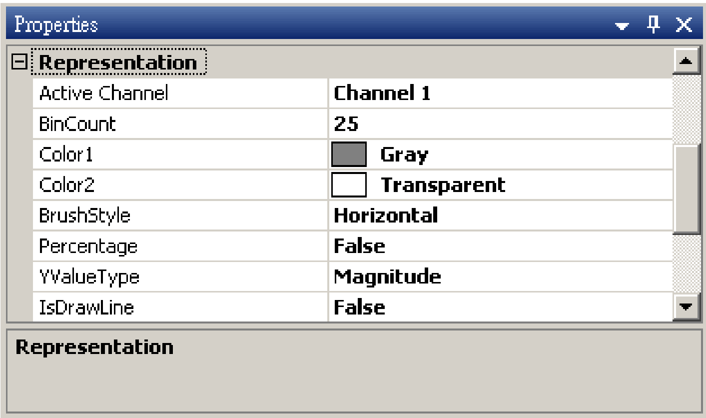
Histogram Viewer is a graphical display SFO which will draw rectangular bars on the graph. It is included in the DataDemon Professional.

Introduction

Histogram is used frequently in probability. X-axis represents the category of the data and the y-axis represents the probability of the data and the graph is drawn as rectangular bars.

Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular or Indexed), Audio (which could be real number or complex number, single channel or multi-channel, Regular). Histogram Viewer and Channel Viewer are very similar; however, there are more variable options for Histogram Viewer.



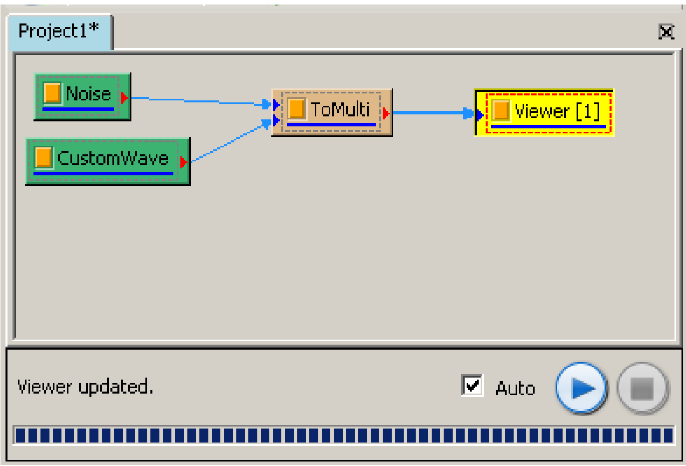
Property Name	Property Definition	Default Value
Active Channel	Specify the Channel to be drawn when there are multiple channels connecting to the viewer (**NOTE:** This option will only appear if there is a multi-channel data connecting to it).	Channel 1
Bincount	Set the number of rectangular bars on the graph.	25
Color1	Set the first color of the bar.	Grey

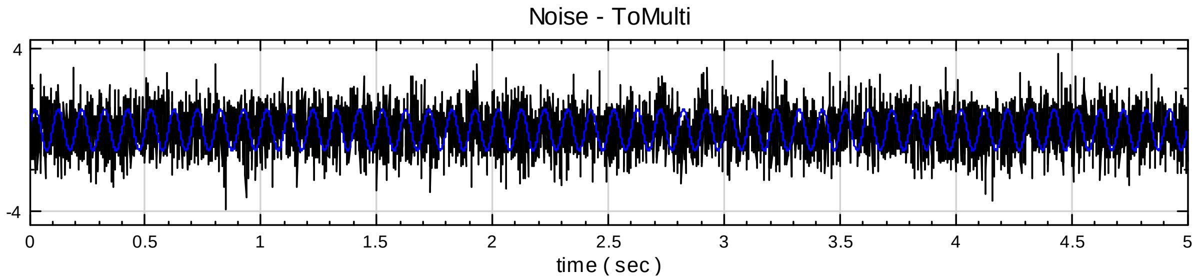
Color2	Set the second color of the bar.	Transparent
BrushStyle	Set the direction of the brush stroke for the color on the bar.	Horizontal
PercentStyle	Select True to change the y-axis to percentage.	False
YValueType	Select different ways to display the y-axis from a selection of Magnitude, Phase, Real, Imagine, Gain and Powerspectrum. Normally this option is used for spectrum data. When there are multiple channels in the signal. (**TIPS:** Please look up Chapter 4.4 Map to Real for more information).	Magnitude
IsDrawLine	Select True to draw lines connecting the top of the rectangular bars together.	False

Example

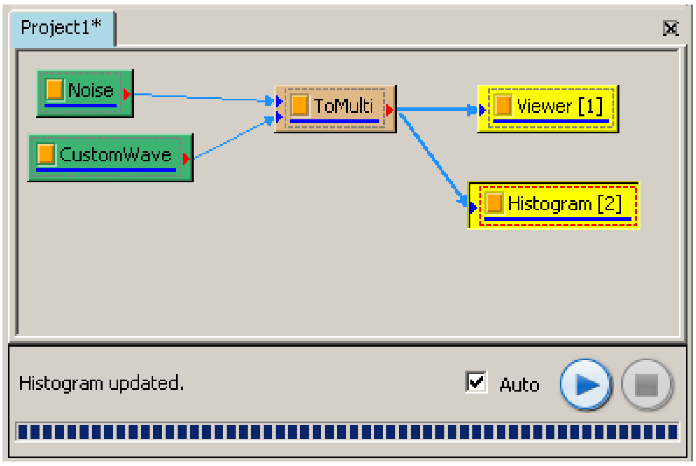
Create a Gaussian Noise and a CustomWave and use *Merge to Multi-channel* to turn both signals into one multi-channel signal and output it to a Box Plot Viewer.

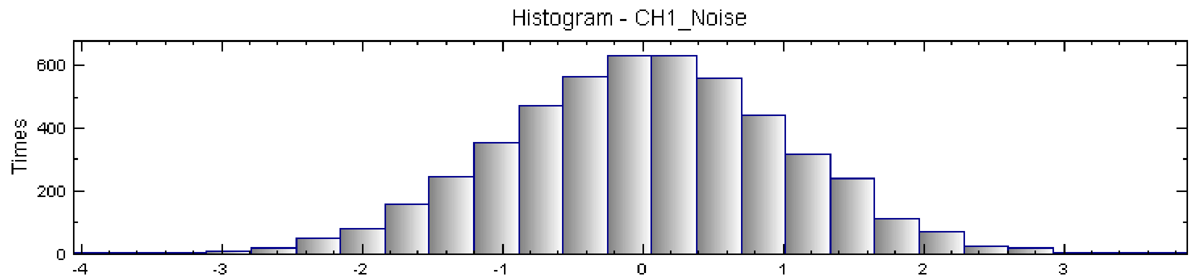
1. Create *Source→Noise* and change its *Properties/Noise/NoiseType* to Gaussian and set the *Properties/Source/TimeLength* to 5. Now create *Source→Custom Wave* and change its *Properties/Source/TimeLength* to 5. Connect both signal waves into *Conversion→Merge To Multi-Channel* and then output it to *Viewer→ Channel Viewer*.



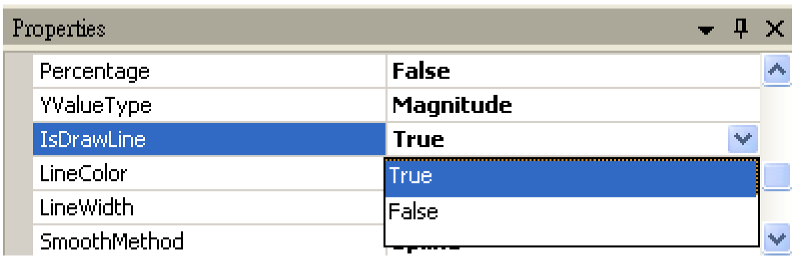


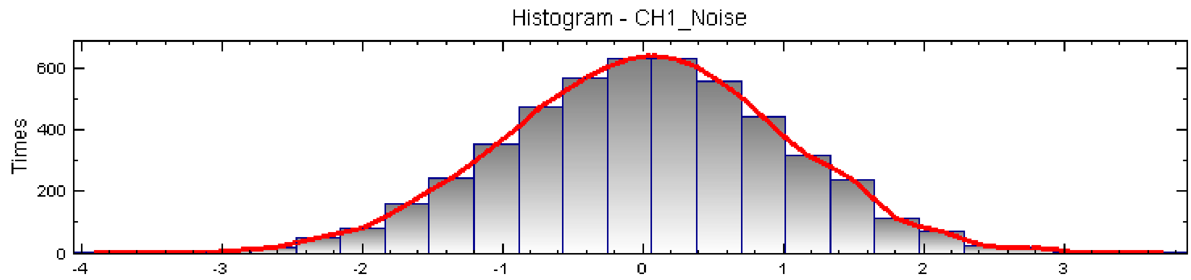
2. Connect ToMulti SFO to *Viewer→Histogram Viewer* and the graph of Channel 1 will be displayed.



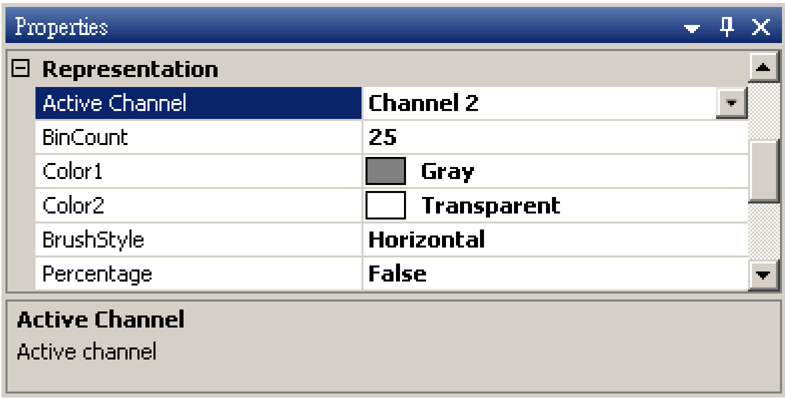


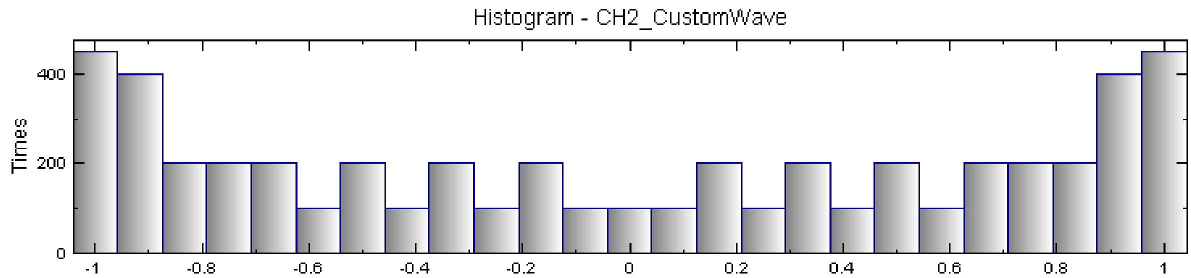
3. Edit the *Properties/Representation/IsDrawLine* to True and a red line will be drawn through the top of the rectangular bars.



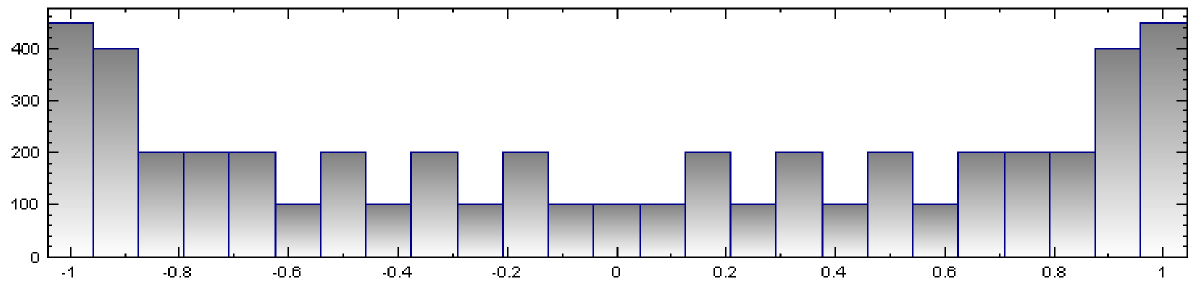


4. Select the *Properties/Representation/Active Channel* to *Channel 2* from the drop-down menu. Now the graph will plot the signal data for Channel 2.





5. Change the *Properties/Representation/BrushStyle* to *Vertical*, and the colors on the rectangular bars will now be shaded from top to bottom instead of the default left to right.



Related Functions

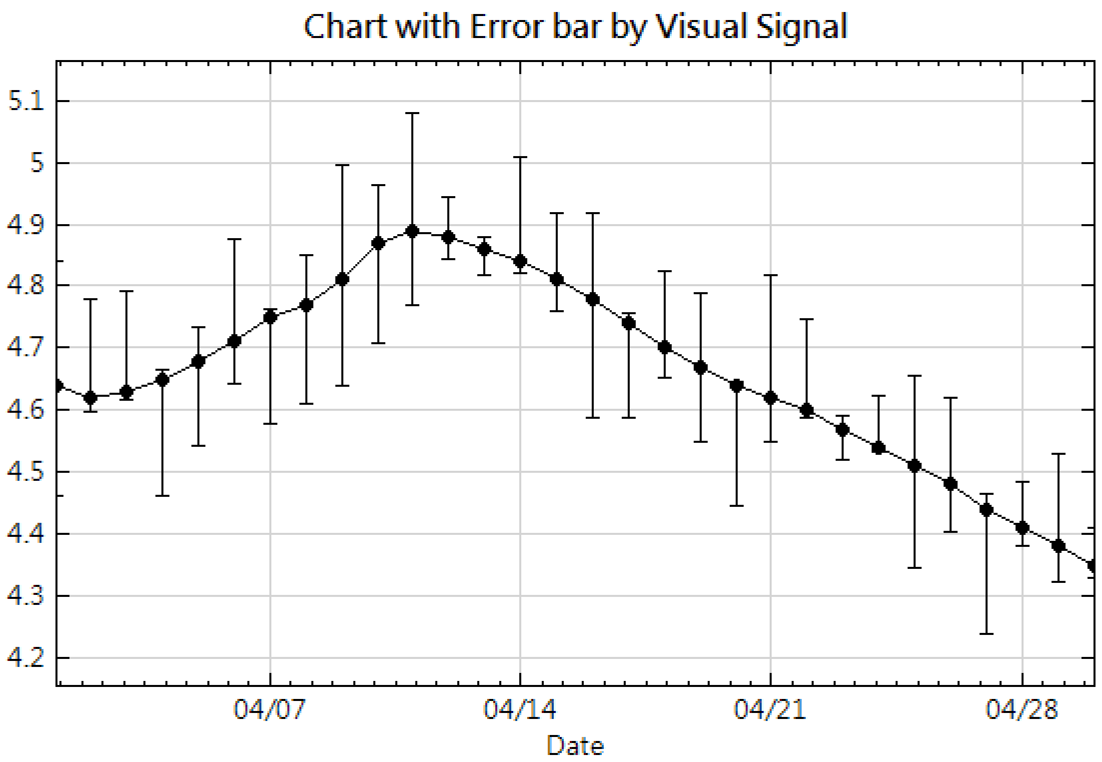
Box Plot Viewer, User Interface, Merge to Multi-Channel and Source.

6.7 Error Bar Viewer (Professional Only)

Error Bar Viewer is used to draw error lines.

Introduction

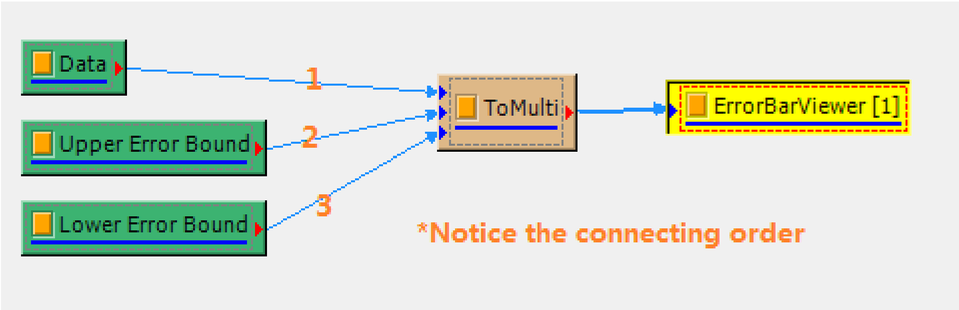
Usually, in the experiment and the measured data will have errors.If you want to show the error in chart to illustrate the reliability or accuracy of the data points, you can use Error Bar Viewer.Error Bar Viewer can plot the typical style as following:



Example

This module requires data ,upper error bound and lower error bound time series must exist sequentially in the same input .

If the three time series are not in the same input, you can use the module 'Merge to Multichannel' to merge the time series into the same input, as following:



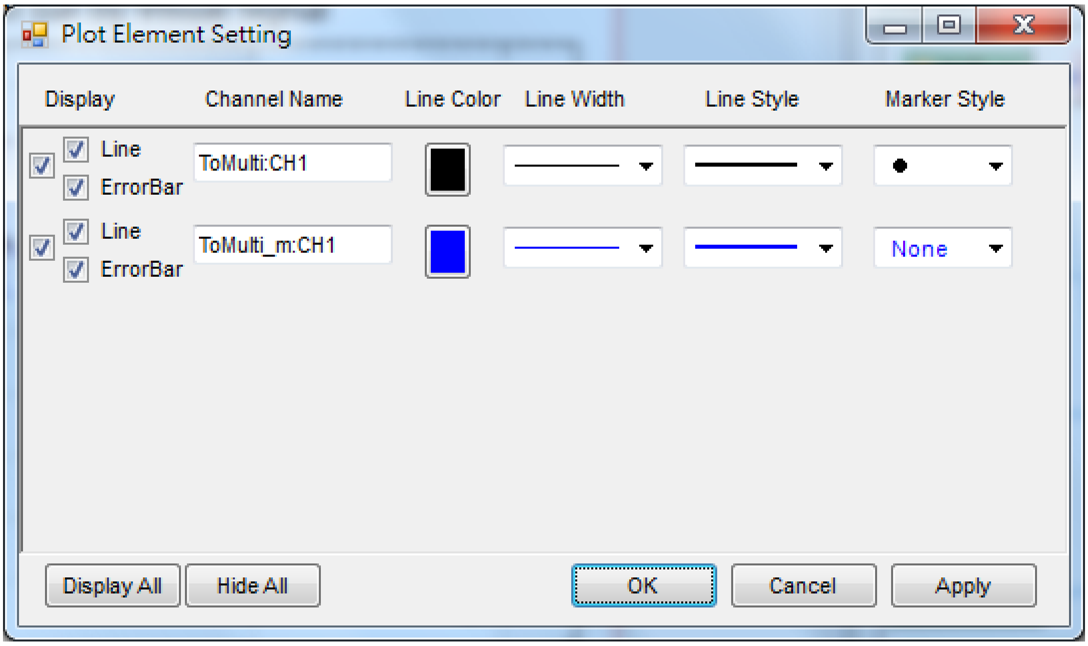
Properties

This module accepts input of Multiple-Channel-Regular Signal and Audio.

Notice that error message will show if the channel number of the input are not equal to 3.Other adjustable parameters are Appearance、Channel、Font and Colors、Module、Representation、Module、Representation、Title.They are almost the same like the parameters of other Viewer,only Error Bar Plot Element Editor under Representation is unique to this module.

Appearance	
Channel	
Fonts and Colors	
Grid	
Module	
Representation	
TimeUint	day
LegendPosition	None
AutoLegendNames	False
XAxisType	LinearAxis
Error Bar Plot Elem Editor	PlotEditor
DataValueType	Magnitude
Hold Plot Range	False
XMin	auto (2008/04/01 00:00:00)
XMax	auto (2008/04/30 00:00:00)
YMin	auto (4.15616313190199)
YMax	auto (5.16524462935515)
DateTime Format	Auto
Show Title	True
Show XAxis	True
Show YAxis	True
Title	

Open Error Bar Plot Editor, the dialog below will show:



You can choose whether to display error bars in the figure,name for channel,adjust marker style of the data point and so on.Currently the color of error bar and line can not be set separately.

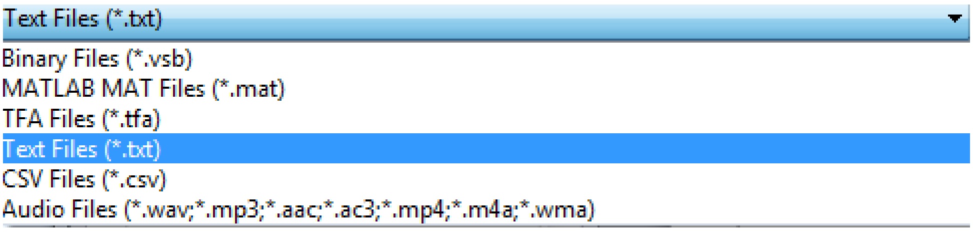
Chapter 7: Writer Signal Flow Object

7.1 Write Data & Export to Excel

The Write Data and Export to Excel functions allows you to export or save DataDemon information into numerous types of file formats.

Properties

Write Data can save data to six different file types: MATLAB MAT Files(*.mat), TFA Files(*.tfa), Text Files(*.txt), CSV Files(*.csv), Wave Files(*.wav),Binary Files(*.vsb).

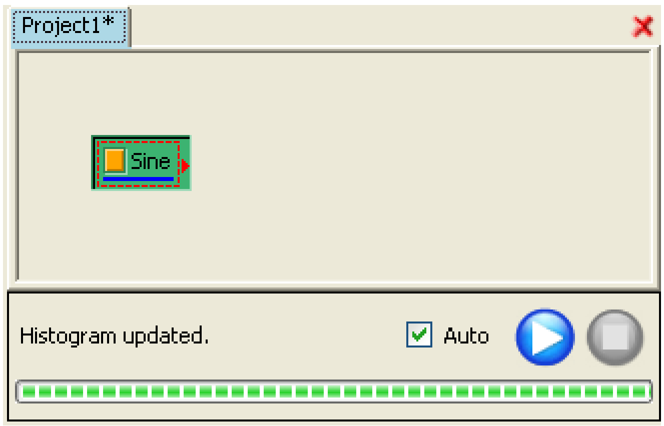


Export to Excel will export the data into Excel. The first column will be the X Values and from the second columns onwards represents the number of channels. Each row in the file contains the data values of the signal. Both Write Data and *Export to Excel* can save the information created by all Signal Flow Objects except Viewer SFOs and Annotation SFOs.

Example

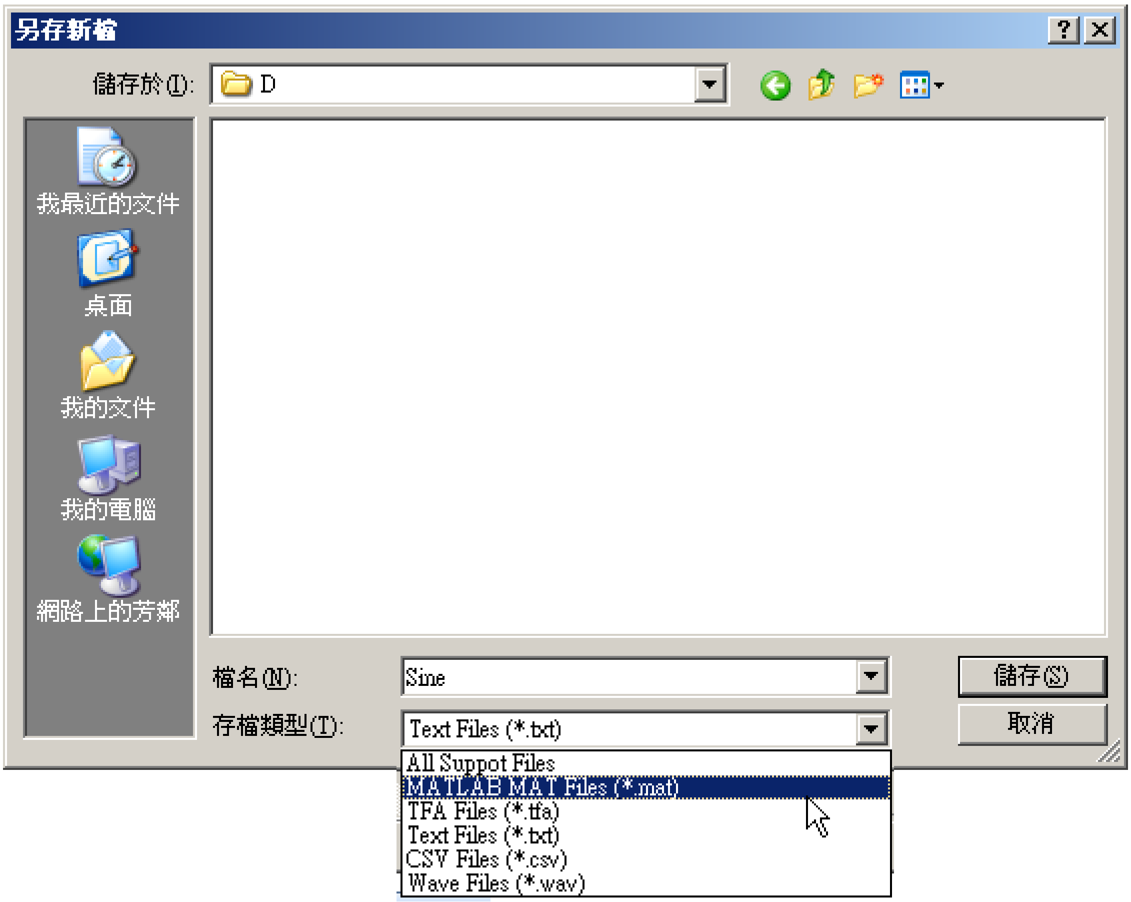
In the following examples, demonstrations are given on how to save a real signal, spectrum signal, spectra signal and numeric signal to a file.

1. Real Signal



Click on the Sine Wave SFO on the *Network Workspace* and then click on the *Save Data to File* button the *Netowork Window Toolbar* to save the information.

After clicking on the *Save Data to File* button a Save As window will appear, allowing the user to save different types of file formats.



If the information was to be saved as the mat file format, then inside the file will be two variables, from this example it will be Sine_x and Sine_y. Sine_x will store the time and Sine_y will store the data values. The data will be represented in column format. For each additional channel, a new column will represent the data value for the new channel.

If you click on the  *Export to Exce*l button, Excel will automatically open with all the data transferred to the Excel table. X Value column stores the time information of the signal and CH 1 column stores the data values. So if there is more than one channel, e.g. CH2, CH3 etc then each channel will be listed in their own column.

X Value	CH 1
0	-1.8E-16
0.001	0.062791
0.002	0.125333
0.003	0.187381
0.004	0.24869
0.005	0.309017
0.006	0.368125
0.007	0.425779
⋮	⋮

2. Spectrum Signal

Export to Excel and *Write Data to File* on a spectrum signal will result in an output which looks something like this:

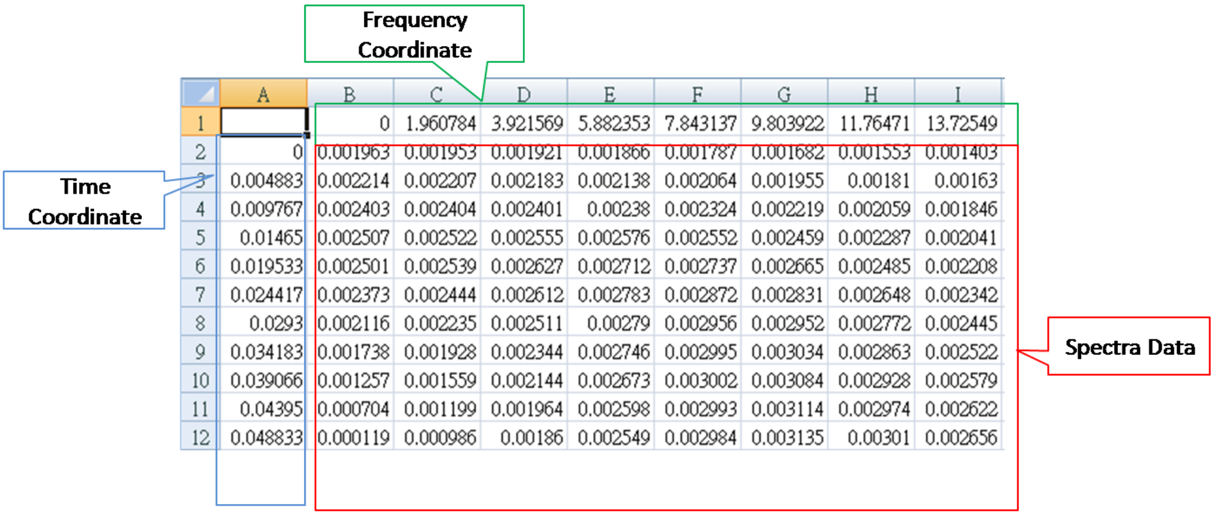
X Value	CH 1 - Real	CH 1 - Imag
0	2.875E-17	0
1	6.319E-07	-0.0002013
2	2.606E-06	-0.0004152
3	6.186E-06	-0.000657
4	1.191E-05	-0.0009488
5	2.084E-05	-0.0013279
6	3.515E-05	-0.0018665
7	5.999E-05	-0.0027303
⋮	⋮	⋮

X Value column stores the time information of the signal and CH 1 –Real column stores the real data values, CH1 –Imag column stores the imaginary part and if there are more than one channel, the information will be listed in the same way.

DataDemon will view a spectrum signal like a multi-channel signal. X Value will store the frequency and the rest will be the same as above.

3. spectra

Export to Excel with a Spectra signal will result in something like this:



	A	B	C	D	E	F	G	H	I
1		0	1.960784	3.921569	5.882353	7.843137	9.803922	11.76471	13.72549
2	0	0.001963	0.001953	0.001921	0.001866	0.001787	0.001682	0.001553	0.001403
3	0.004883	0.002214	0.002207	0.002183	0.002138	0.002064	0.001955	0.00181	0.00163
4	0.009767	0.002403	0.002404	0.002401	0.00238	0.002324	0.002219	0.002059	0.001846
5	0.01465	0.002507	0.002522	0.002555	0.002576	0.002552	0.002459	0.002287	0.002041
6	0.019533	0.002501	0.002539	0.002627	0.002712	0.002737	0.002665	0.002485	0.002208
7	0.024417	0.002373	0.002444	0.002612	0.002783	0.002872	0.002831	0.002648	0.002342
8	0.0293	0.002116	0.002235	0.002511	0.00279	0.002956	0.002952	0.002772	0.002445
9	0.034183	0.001738	0.001928	0.002344	0.002746	0.002995	0.003034	0.002863	0.002522
10	0.039066	0.001257	0.001559	0.002144	0.002673	0.003002	0.003084	0.002928	0.002579
11	0.04395	0.000704	0.001199	0.001964	0.002598	0.002993	0.003114	0.002974	0.002622
12	0.048833	0.000119	0.000986	0.00186	0.002549	0.002984	0.003135	0.00301	0.002656

The first column represents the time, the first row represents the frequency, and the data in-between the first column and the first row represents the signal strength.

Related Functions

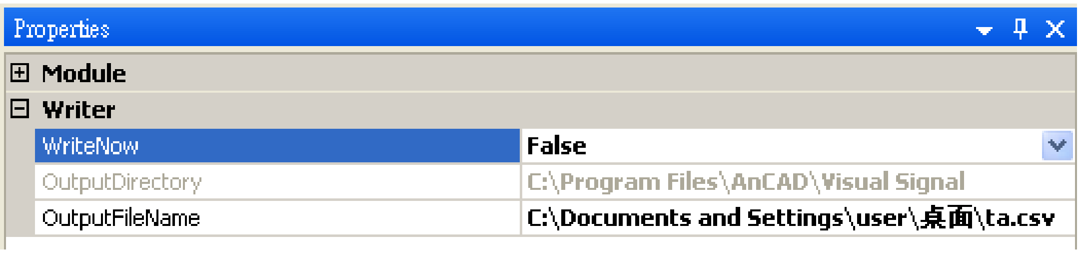
Writer.

7.2 csv Writer

Export the data to csv format (the data is separated by commas).

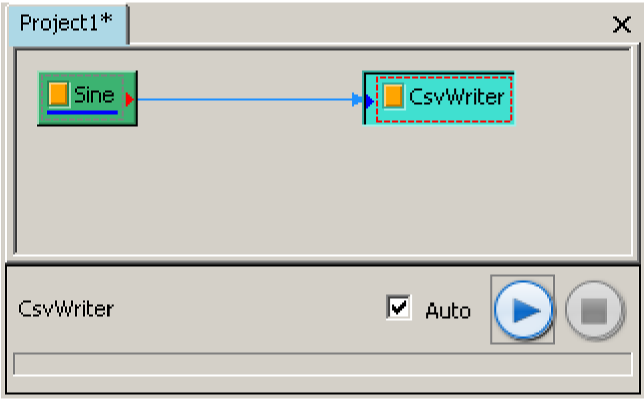
Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular), Audio (which could be real number or complex number, single channel or multi-channel, Regular).

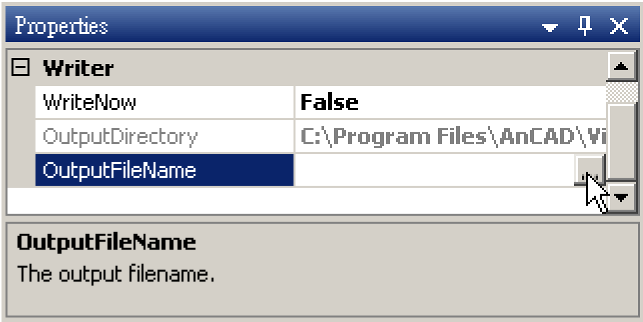


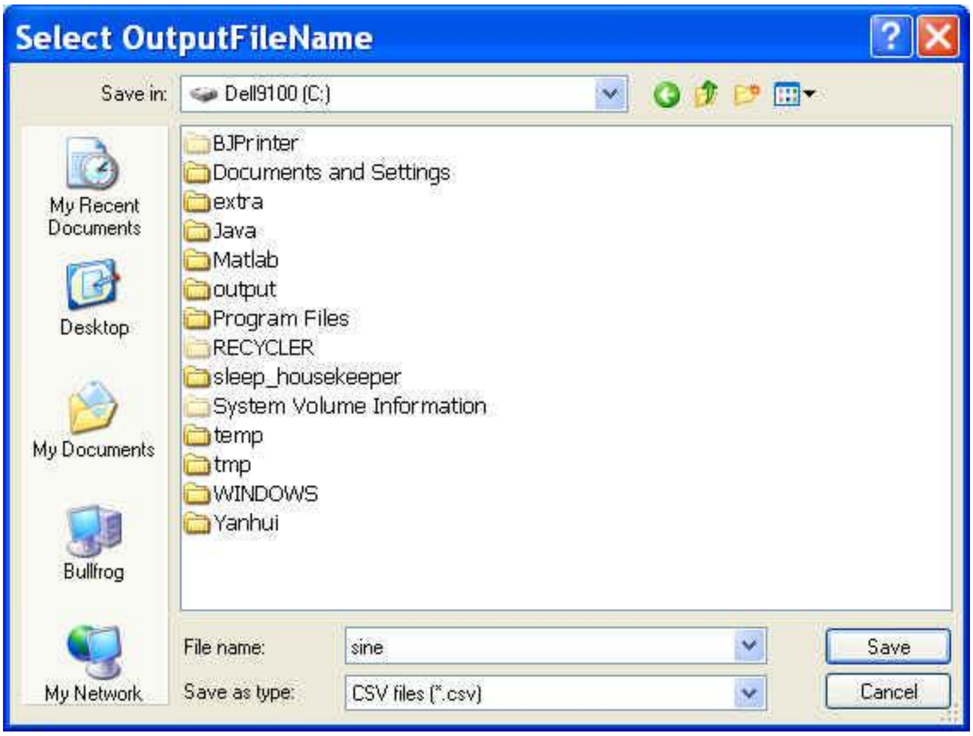
Property Name	Property Definition	Default Value
WriteNow	Select True to write the data to file.	False
OutputDictory	Displays the default location of the files to be saved (**Note:** You can edit the default location from the main menu under *Tools* → *Preference*).	
OutputFileName	Select the location to save the file (**NOTE:** With the file name and location entered, the file is only saved when WriteNow is set as True).	

Example

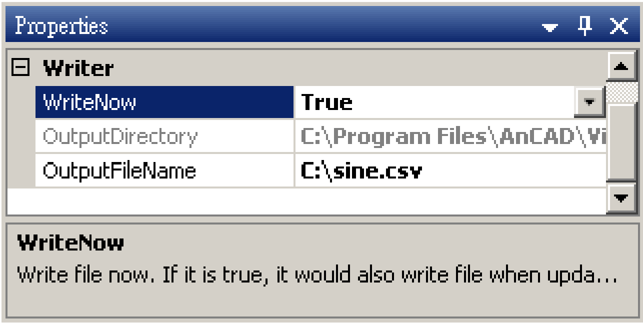


Create a Source→Sine Wave SFO and connect it to *Writer→CsvWriter* to save the data to csv format. Click on the *Properties/Writer/OutputFileName* field and a button will appear at the right hand side of the field. Click on the button to enter the name for the file and the location to save the file. Enter the file name as “sine” and save the file to the location C:\ directory.





Since *Properties/Writer/WriteNow* is set as *False*, the file has not be saved yet. Now change the WriteNow option to *True* and the file will be saved in C:\ directory.



Related Functions

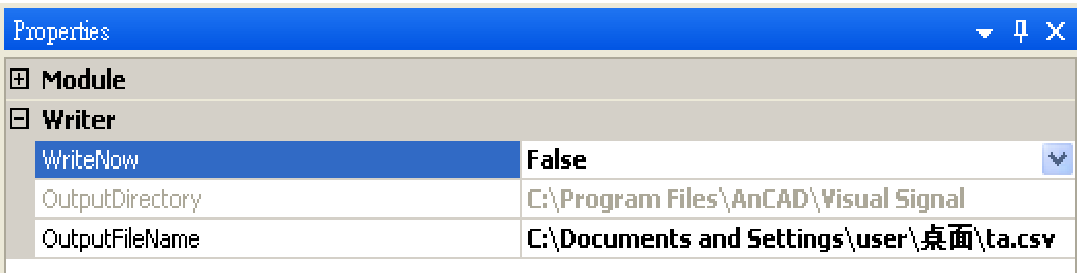
Text Writer, TFA Writer, Wave Writer.

7.3 Text Writer

Turns signal data and audio data to text file.

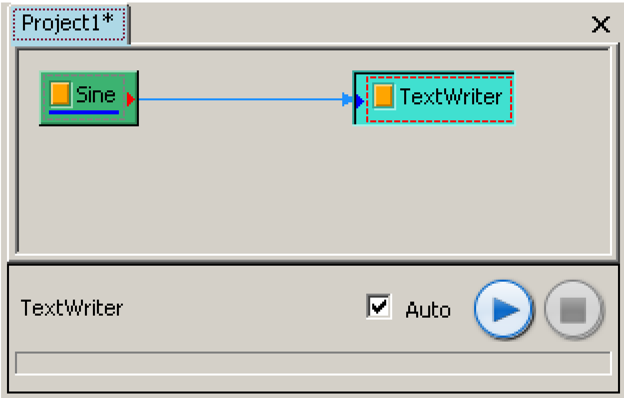
Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular), Audio (which could be real number or complex number, single channel or multi-channel, Regular).

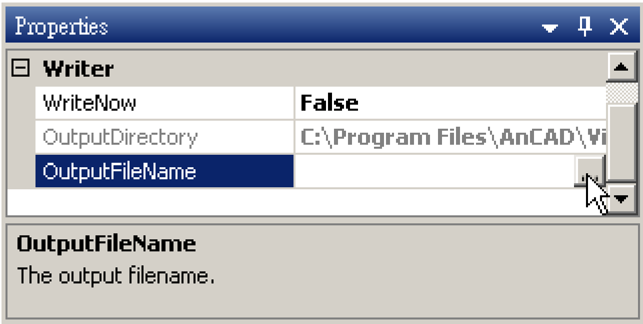


Property Name	Property Definition	Default Value
WriteNow	Select True to write the data to file.	False
OutputDictory	Displays the default location of the files to be saved (**Note:** You can edit the default location from the main menu under *Tools* → *Preference*).	
OutputFileName	Select the location to save the file (**NOTE:** With the file name and location entered, the file is only saved when WriteNow is set as True).	

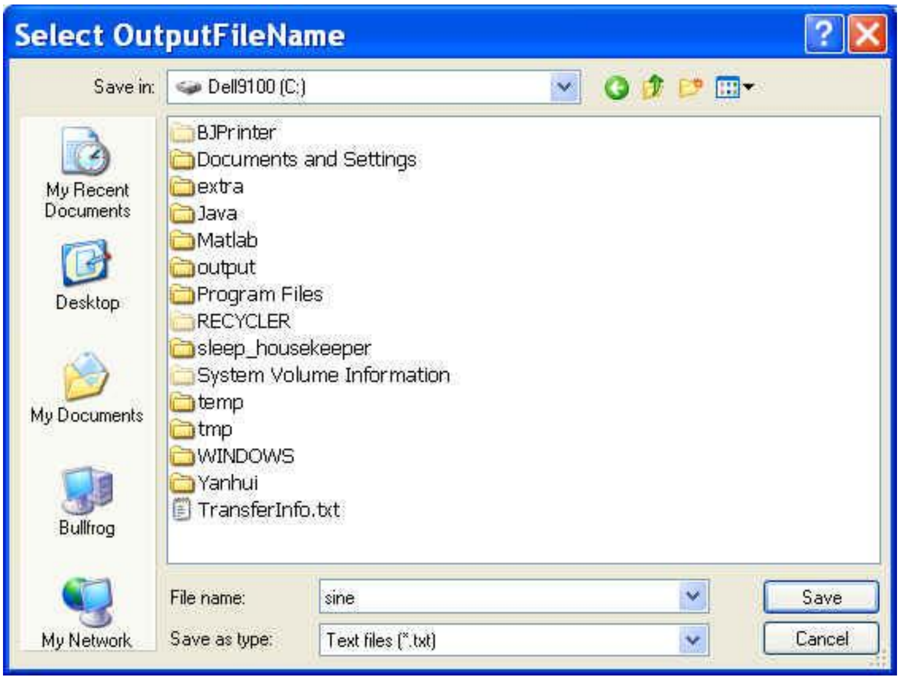
Example

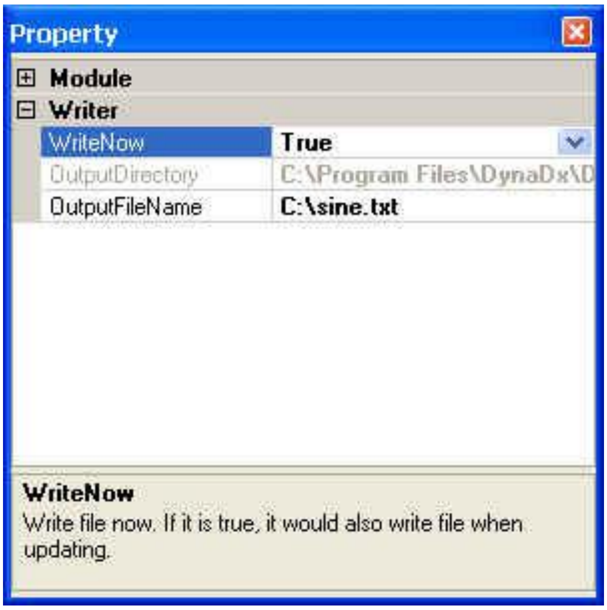


Output a Sine Wave signal to a txt file. Create *Source→Sine Wave* to *Writer→Text Writer* to save the data into a text format. Click on the *Properties/Writer/OutputFileName* field and a button will appear at the right hand side of the field. Click on the button to enter the name for the file and the location to save the file.



Enter the file name as "sine" and save the file to the location C:\ directory. Since *Properties/Writer/WriteNow* is set as *False*, the file will not be saved yet. Now change the WriteNow option to *True* and the file will be saved in C:\ directory.





Related Functions

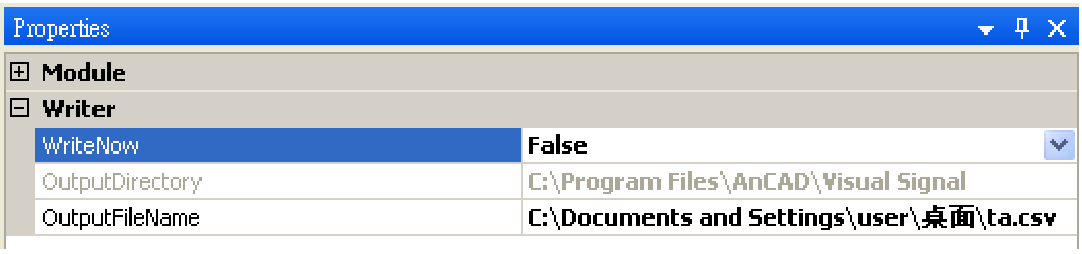
Csv Writer, TFA Writer, Wave Writer and Text Writer

7.4 TFA Writer

Export multi-channel data into .tfa format.

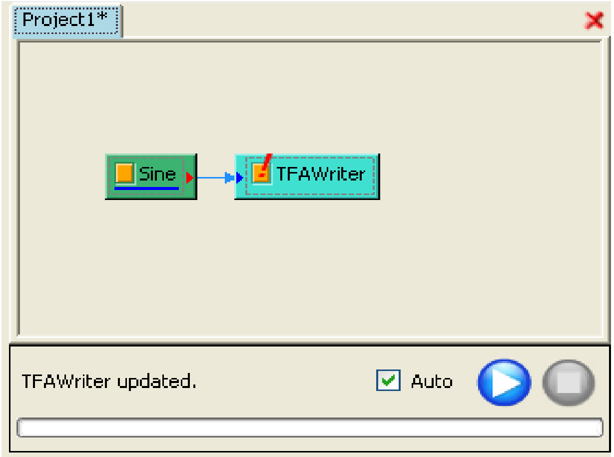
Properties

This module accepts input of Signal (which could be real number or complex number, single channel or multi-channel, Regular), Audio (which could be real number or complex number, single channel or multi-channel, Regular).

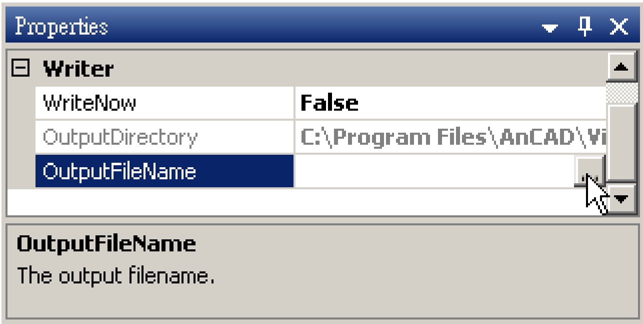


Property Name	**Property Definition**	**Default Value**
WriteNow	Select True to write the data to file.	False
OutputDictory	Displays the default location of the files to be saved (**Note:** You can edit the default location from the main menu under *Tools* → *Preference*).	
OutputFileName	Select the location to save the file (**NOTE:** With the file name and location entered, the file is only saved when WriteNow is set as True).	

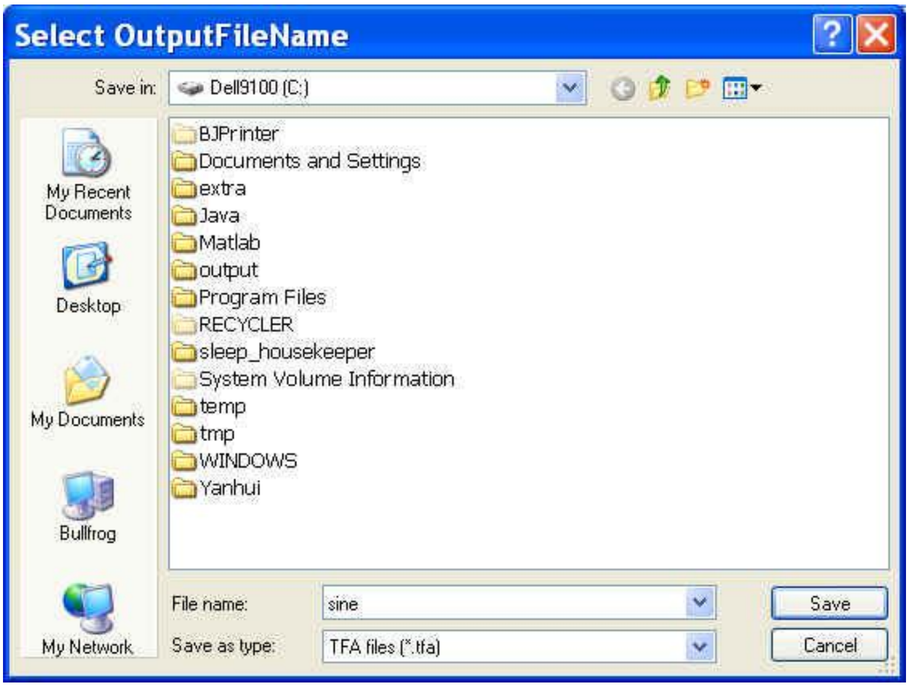
Example

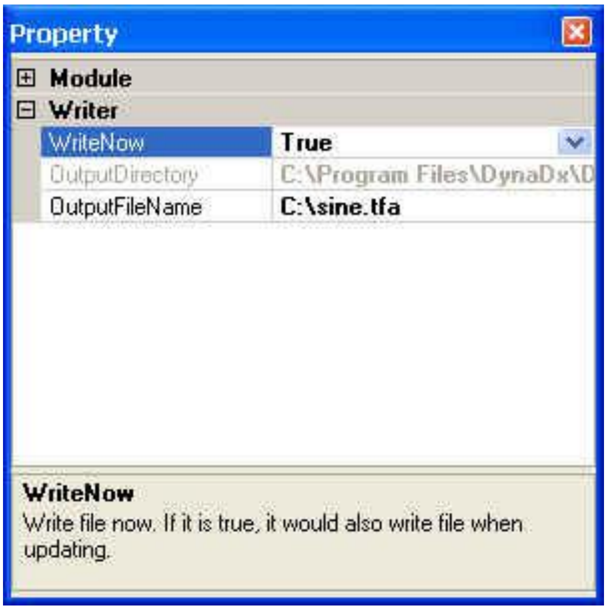


Output a Sine Wave signal to a tfa file. Create a *Source→Sine Wave* and connect it to a *Writer→TFA Writer* to save the data into a tfa format. Click on the *Properties/Writer/OutputFileName* field and a button appears at the right hand side of the field. Enter the file name and the file will be stored in the OutputDirectory folder.



Enter the file name as "sine" and save the file to the location C:\ directory. Since *Properties/Writer/WriteNow* is set as False, the file has not be saved yet. Now change the WriteNow option to True and the file will be saved in C:\ directory.





Related Functions

Csv Writer, TFA Writer, Wave Writer.

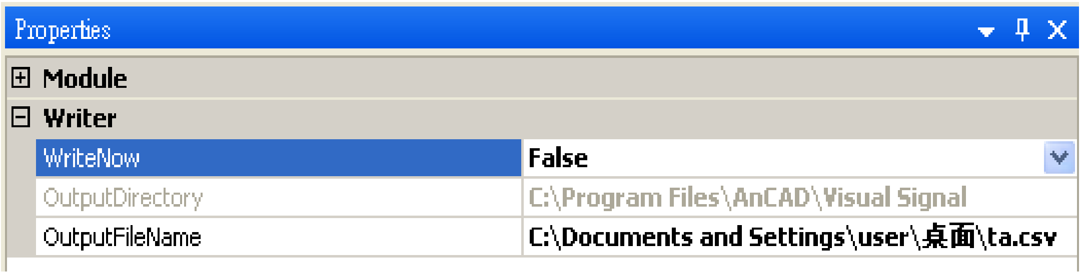
7.5 Wave Writer

Export a signal or an audio into a wave file format.

Properties

This module accepts input of Audio (which could be real number, single channel or multi-channel, Regular).

The default output for WriteNow is set as *False* so the wav file will not be exported to a file. Only when this property is set as *True* then the file will be exported to a target location (default location).

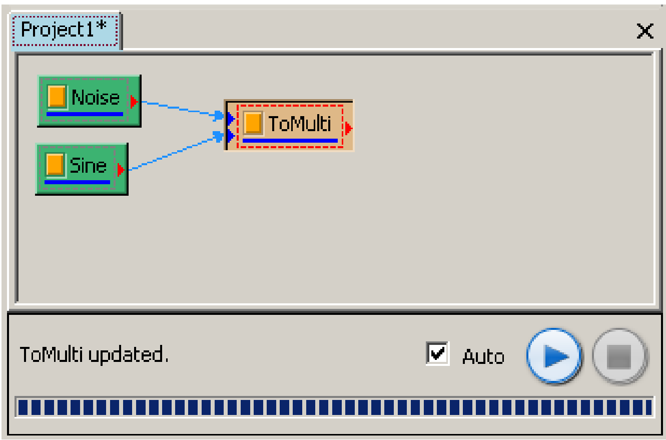


Property Name	Property Definition	Default Value
Play	Set True to play the audio file after the file has been saved or play the audio file right away if the file is already saved with the default audio player of the computer.	False
WriteNow	Select True to write the data to file.	False
OutputDictory	Displays the default location of the files to be saved (**Note:** You can edit the default location from the main menu under *Tools* → *Preference*).	
OutputFileName	Select the location to save the file (**NOTE:** With the file name and location entered, the file will only save when WriteNow is set as True).	

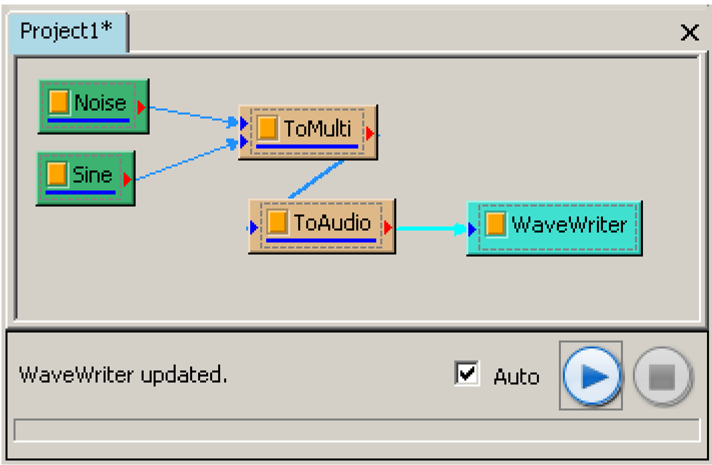
Example

In this example, create a noise wave and a sine wave and combine them to become a multi-channel data then convert the multi-channel data to audio.

1. Create *Source→Noise Wave* with *Properties/Noise/NoiseType* set to White Noise, *Properties/Source/SamplingFreq* set to 1000 Hz, and set the *Properties/Source/TimeLength* to 2 seconds. Then create *Source→Sine* with the same settings, SamplingFreq set as 1000, and TimeLength set as 2. And connect both signal data to a *Compute→Conversion→Merge* to Multi-channel SFO.



2. Connect the ToMulti SFO to *Conversion→Convert to Audio* to turn the signal data to audio. Set the Sample Rate to 1000 Hz, Bits per sample to 16 bps so that once it is converted to audio it will be accepted by *Writer→Wave Writer*.



Related Functions

Noise, Sine, Merge to Multi-channel and Convert To Audio.

Chapter 8: Macro and Container Signal Flow Objects

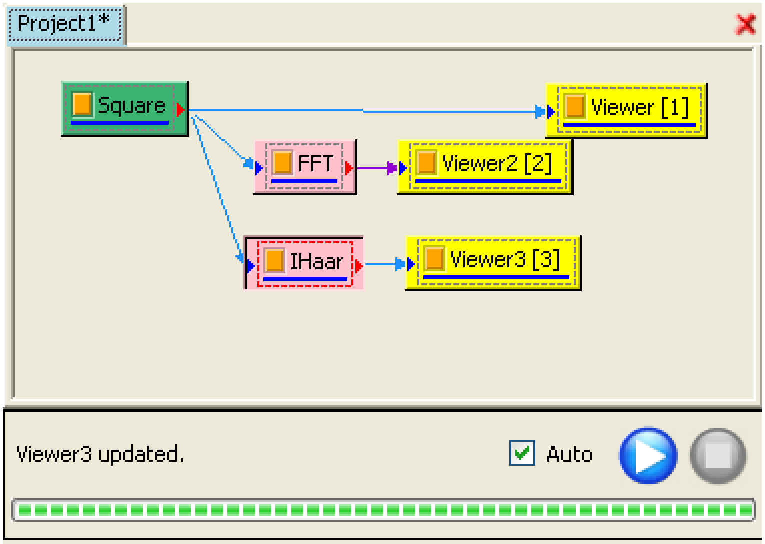
8.1 Macro (Professional Only)

Macro allows user to save all current Signal Flow Objects (SFOs) and their network relationship with each other into a macro file, so that the same SFOs can be recreated in other projects by opening the saved macro file. By utilizing this feature available in the professional version, users can save a lot of time without repeating the same process of recreating the same SFOs over and over again.

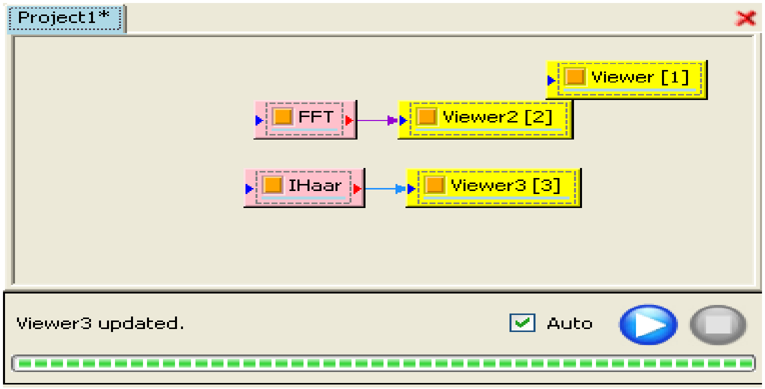
Example

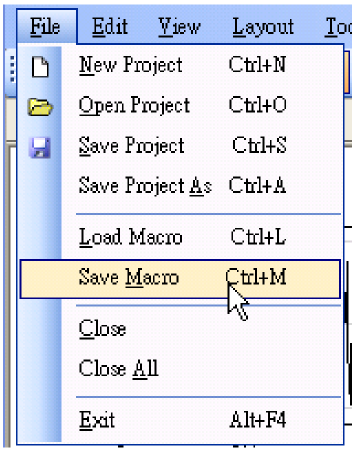
Evaluating a Noise Wave.

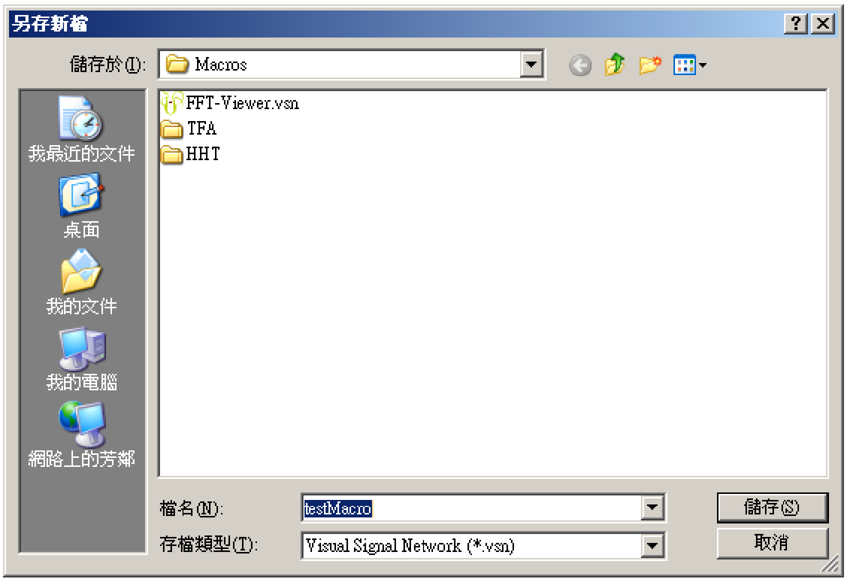
1. The Signal Flow Diagram between the SFOs is shown below.



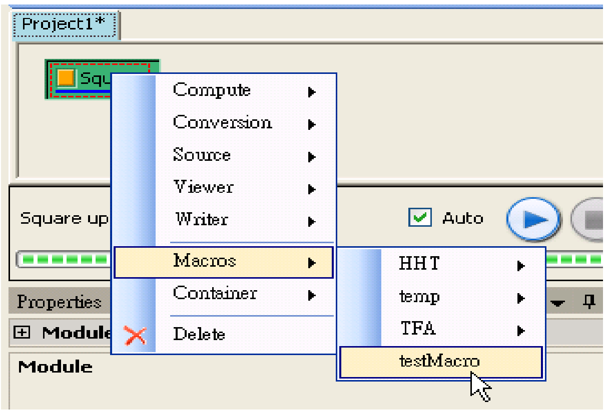
2. The user can save the current SFOs setup into a macro file without the need to save the project. Go to the main menu and click on *File* and then click on *Save Macro* to save it to a file. In the example, the macro file is saved as *testMacro* (**TIPS:** It will be easier to locate macros if you save macros into the Macro folder directory or save the file with a *macro* prefix or suffix in its file name).







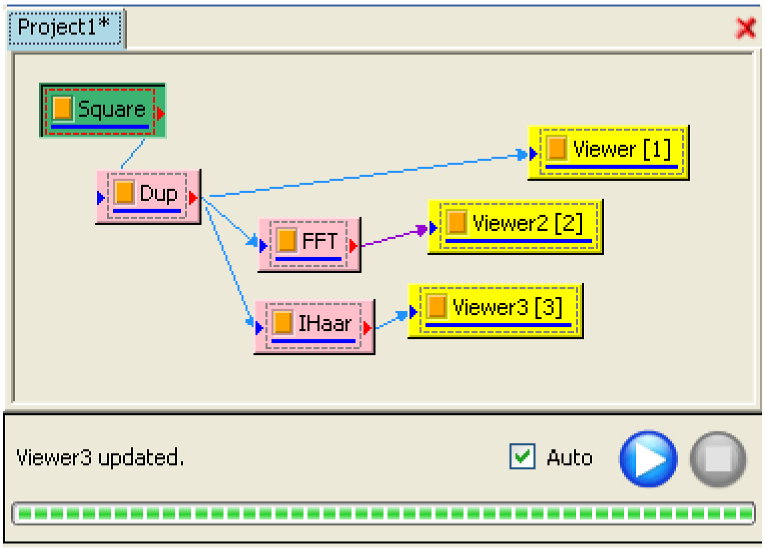
3. Create a new project to test out the macro saved previously. Create a *Source→ Square Wave* and right click on the Square Wave SFO to open up the *Network Workspace Menu* and select *Macros→testMacro* to load the testMacro file .You can also click on the *File* option on the main menu and select *Load Macro* and choose the macro file to load (**NOTE:** Project files can be loaded as a macros and vice versa, macros can be loaded as projects).



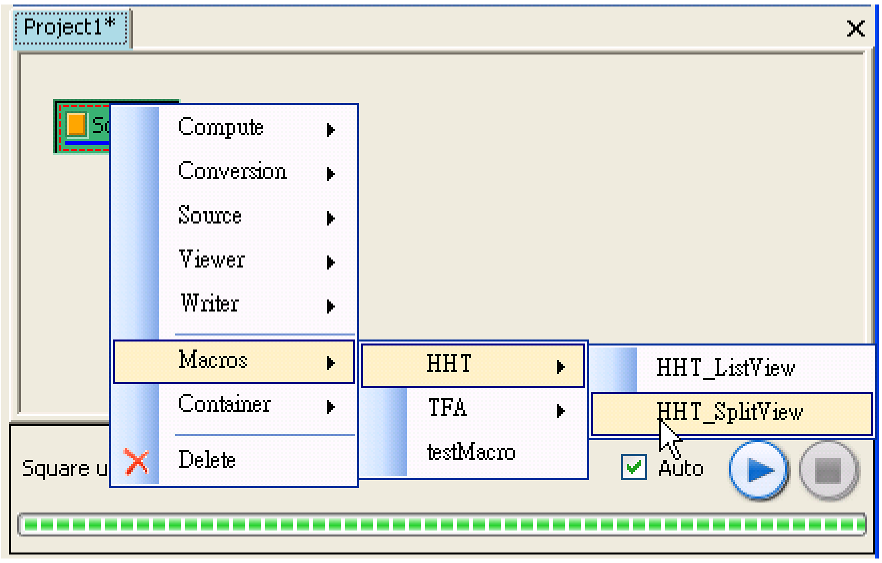
Try to save all macros under the Macro folder directory of DataDemon. So when you try to access the macros from the *Network Workspace Menu*, it will appear in the menu selection under *Macros*.

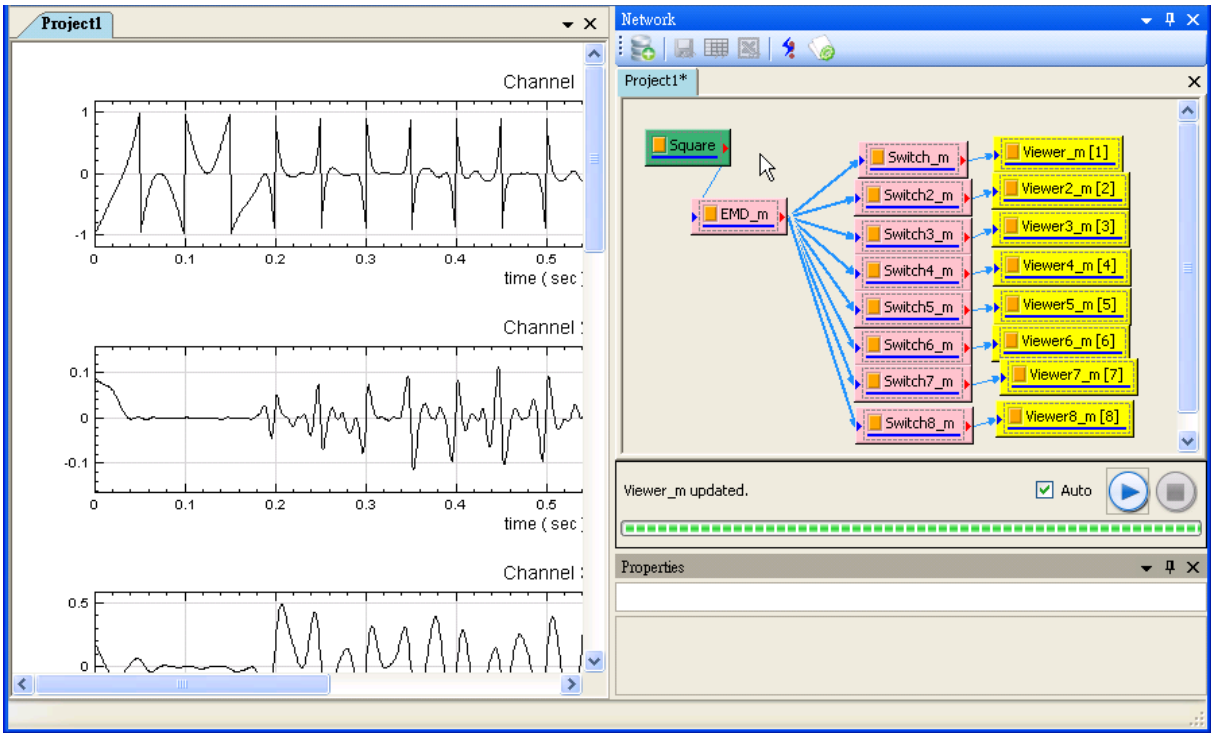
4. When loading a macro into a project, sometimes you will still be required to manually link the source Signal Flow Objects to the rest of the loaded SFOs (**NOTE:** The use of macros is to save the steps in setting up the rest of the Signal Flow Diagram and it is mainly the Source SFOs which are constantly being replaced). To make things easier for yourself, create a Channel→Dup SFO that connects to the rest of the networked SFOs. So the next time you load the macro, just connect the new Source SFO to the Dup SFO and you are ready to view the results without having to drag and connect the new Source SFO to other

SFOs every time.



5. The above macro example is quite simple but most of the time, macros can contain a lot more Signal Flow Objects. In DataDemon there are several predefined macros and one such macro is the HHT_SplitView macro. HHT_SplitView macro will turn a signal into EEMD and calculate and display the graph for each of the channel.





Macro saves time when there is a need to repeat certain processes within different projects. So saving complicated macros and reuse those in different projects can increase work efficiency.

6. Macro file and DataDemon project file are both saved as .vsn file. The only difference is that when you load a .vsn file through *Load Project*, the file will be opened as a project and when you load the same .vsn file through *Load Macro*, the file will be opened as a macro within the project.

Related Functions

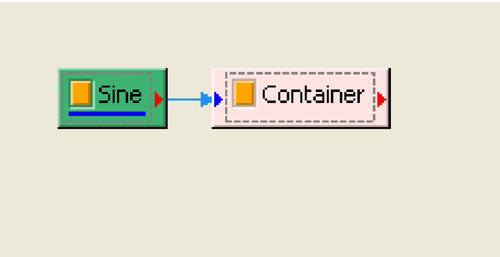
Dup.

8.2 Container (Professional Only)

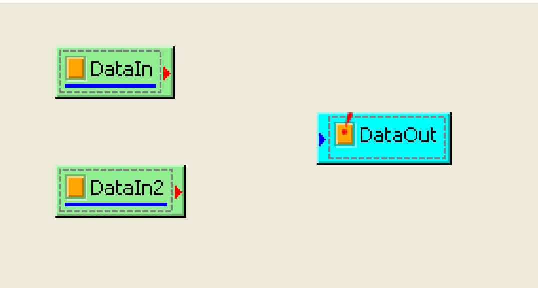
Container allows user to pack SFOs up to an element, so that user can manage programs of complex signal computation.

Introduction

When complex signal anlysis assignments are processing, module elements may fulfill the Network panel window. Now a Container module is added to pack SFOs up into an element.



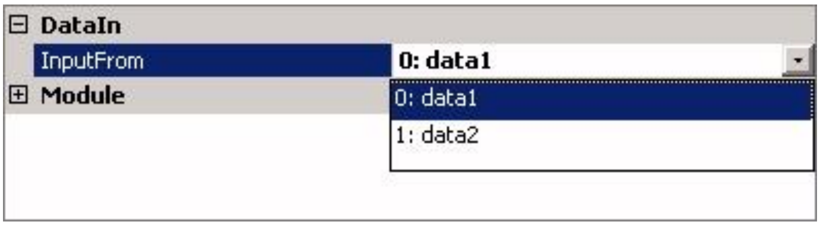
The user can add a Container in the same way of adding a Project, but the compiler of this module belongs to Container so that it can be operated in the compiler area of Container module. Different Containers can have their visual plot areas and input sources (DataIn), but just can own one output (DataOut).



Functions of Container are instructed as below:

NAME	ACTION
Add	Add a Container
DataIn	In Container, gives the data input entry.
DataOut	In Container, gives the data output entry.

When many modules connect to a Container, the user should set the origin of the data by the InputFrom function of DataIn.

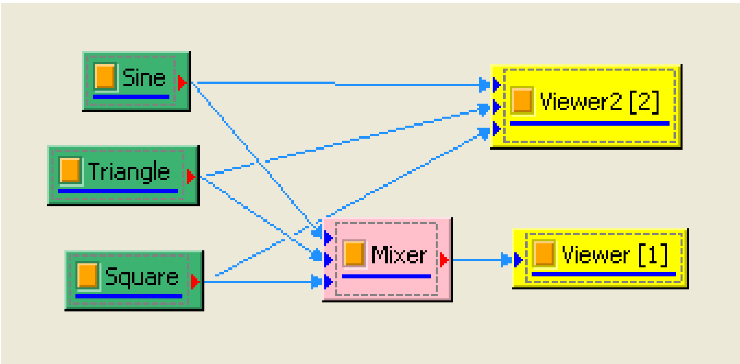


One Container just can own one DataOut.

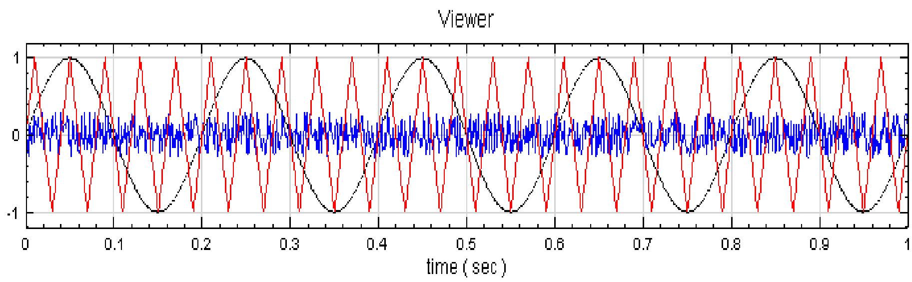
Example

Pack up SFOs into a Container.

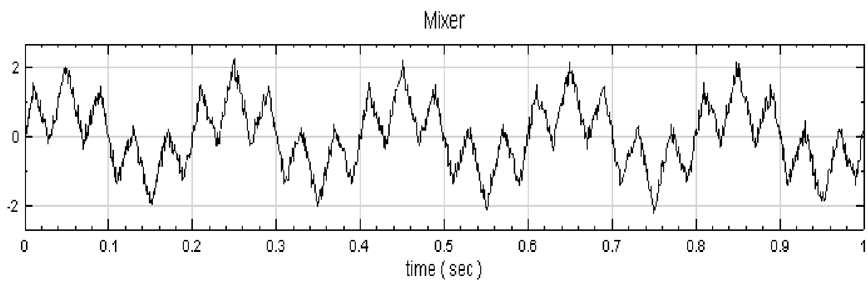
Mix Sine, Triangle and Noise. (For easy viewing, frequencies of the Sine wave and Triangle are different from the Noise.)



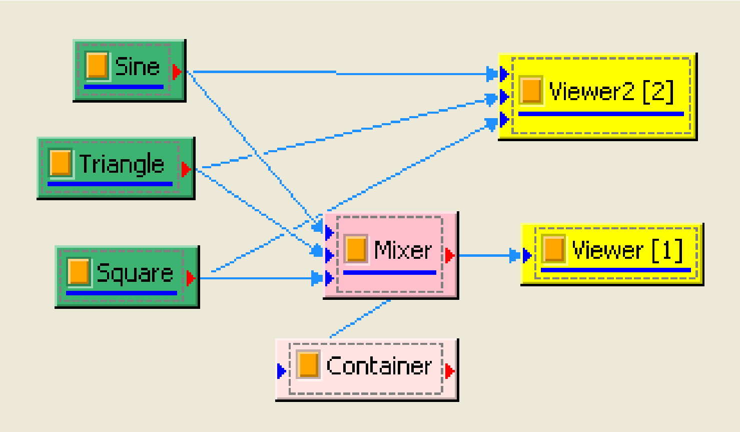
The original signal is:



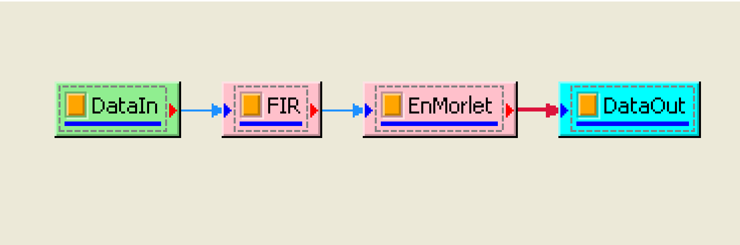
After mixing



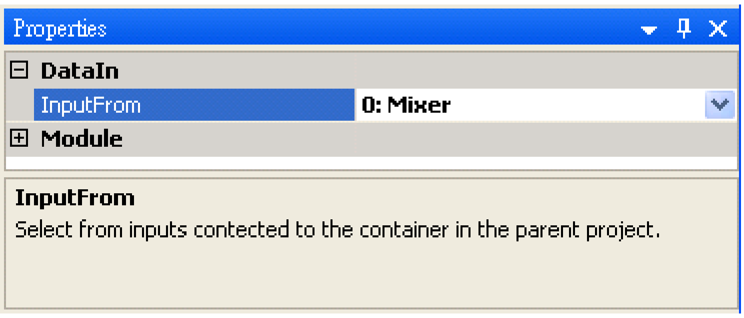
After Mixer, a Container is added by Container / Add. Then the compiler area of Project1.Container is added automatically in the Network Panel. On the left side of the main window, you can find the plot area of the newly added Project1.Container.



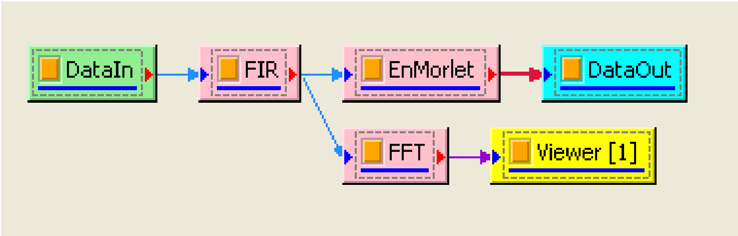
In Container, you need to select Container / DataIn in the context menu and send input data to Container. After DataIn, you connect Compute / Filter / FIR Filter and Compute / TFA / Enhenced Morlet. At the end, Container / DataOut is connected and computation results are sent to Container's ouput entry.



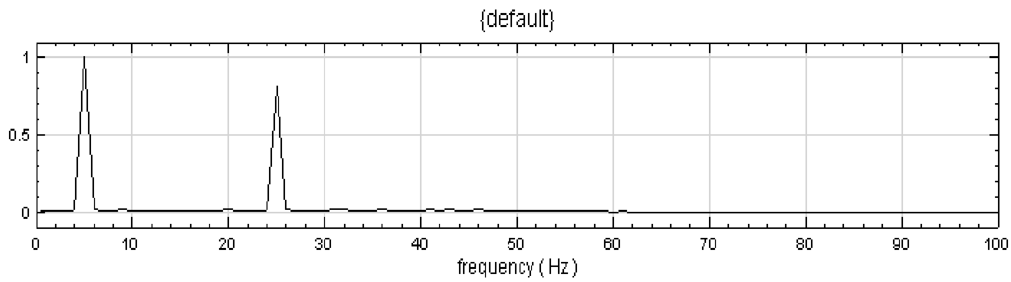
If there are many modules connecting to a Container, the user need to set the origin of data by the parameter named InputFrom in the DataIn. In this simple example, Input is just Mixer.



In addition, the user can add Transform / Fourier Transform after FIR and show the data by Viewer / Channel Viewer.

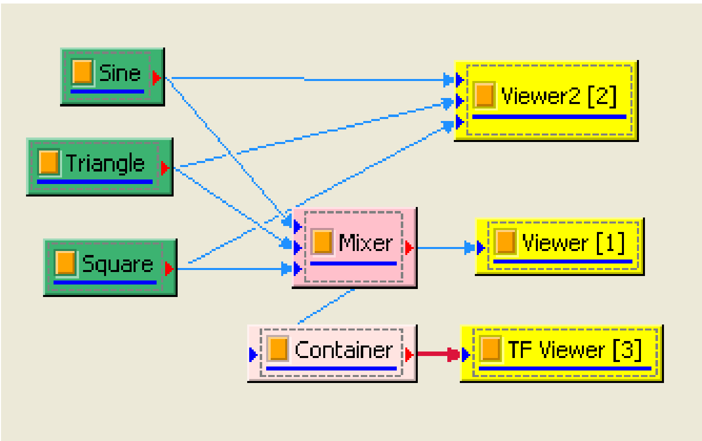


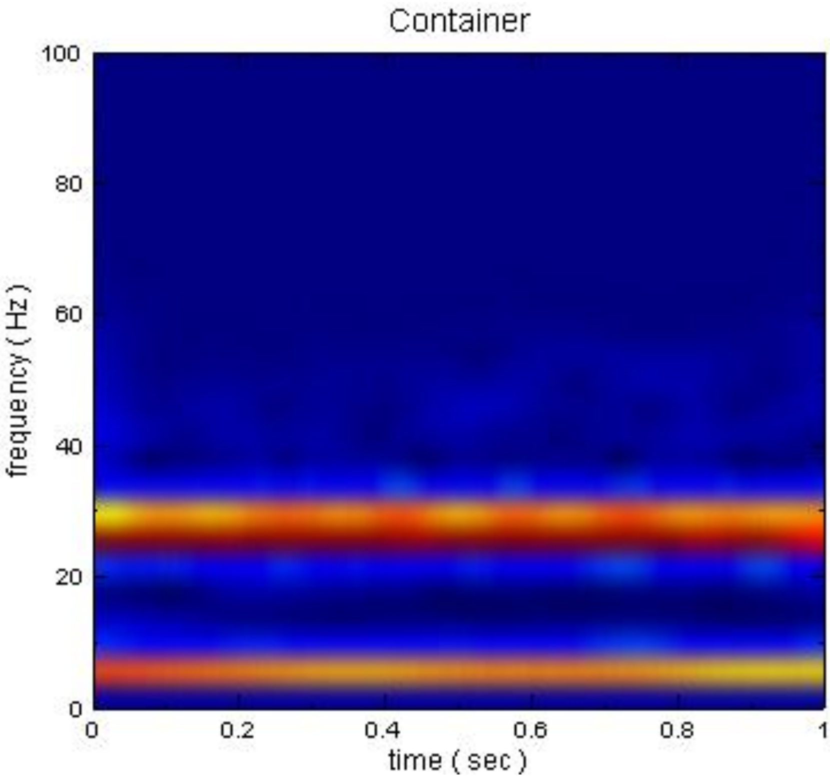
The result of FFT is:



The Viewer of this graph is drawn in the Container.

Return to the original Project, results are obtained by connecting the Container to the TFA Viewer:





Container and Macro are almost the same in concept, so two points need to be clarified: unlike the Macro, Container can not exist alone. It just can be saved with the Project which it belongs to. In Container, the Macro can be read and operated the same way as in Project.

Related Functions

Macro